(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 296 781 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 88305609.5

(22) Date of filing: 20.06.88

(51) Int. Cl.4: G01N 30/86 , G01N 30/88 , C07K 1/00

(30) Priority: 19.06.87 US 65012

(43) Date of publication of application:
28.12.88 Bulletin 88/52

(84) Designated Contracting States:
DE FR GB NL SE

(71) Applicant: APPLIED BIOSYSTEMS, INC.
850 Lincoln Centre Drive
Foster City California 94404(US)

(72) Inventor: Hunkapiller, Michael W.
1333 Pebble Drive
San Carlos California 94070(US)

(74) Representative: West, Alan Harry et al
R.G.C. Jenkins & Co. 26 Caxton Street
London SW1H 0RJ(GB)

(54) Quantitation of chromatographic information.

(57) A process and apparatus are disclosed for quantitating chromatographic information. The method uses a discrete, linear, translation invariant filter function a_N^α , where N is a measure of the filter function width and α is a parameter whose value determines signal to noise characteristics of the filter function. First a chromatographic analysis of a sample is performed to obtain a first chromatogram. Then the first chromatogram is filtered with the filter function. with N set to approximate the width of peaks obtained in the first chromatogram, and α is set to filter out high frequency noise from the first chromatogram to obtain a second chromatogram having a first filtered baseline. The second chromatogram is then filtered with α set to resolution enhance peaks in the second chromatogram to obtain a third chromatogram having a baseline which is substantially the same as the first filtered baseline. The second chromatogram is then subtracted from the third chromatogram to obtain a fourth chromatogram which is baseline corrected.

QUANTITATION OF CHROMATOGRAPHIC INFORMATION

Background of the Invention

This invention relates to a method and apparatus of general utility for analyzing chromatographic information, and particularly to the use of such for analyzing chromatographic information pertaining to the automatic determination of peptide sequences.

The chemical process employed by protein/peptide sequencers is derived from a technique originated by Pehr Edman in the 1950s for the sequential degradation of peptide chains (Edman, Acta Chem. Scand. 4, 283, 1950; Edman and Begg, Eur. J. Biochem. 1, 80, 1967). The first step in this degradation is selective coupling of a peptide's amino-terminal amino acid with the Edman reagent, phenylisothiocyanate (PITC), a reaction catalyzed by an organic base delivered with the coupling reagent. The second step is cleavage of this derivatized amino acid from the remainder of the peptide, a reaction effected by treating the peptide with a strong organic acid. Each repeated coupling/cleavage cycle occurs at the newly-formed amino-terminal amino acid left by the previous cycle. Thus, repetitive cycles provide sequential separation of the amino acids which form the primary structure of the peptide.

The sequencing process is not completed by the Edman degradation alone. Once the amino acids are removed from the sample, they must be analyzed to determine their identity. Since the cleaved amino acid derivative, the anilinothiazolinone (ATZ), is not generally suitable for analysis, it is converted to the more stable phenylthiohydantoin (PTH) form before analysis is attempted. In modern sequencers (Wittmann-Liebold et al Anal. Biochem 75, 621, 1976; Hewick et al J. Biol. Chem. 256, 7990, 1981), this conversion is accomplished automatically in a reaction vessel separate from that in which the Edman degradation occurs. The ATZ produced at each degradation cycle is extracted from the peptide with an organic solvent, transferred to the reaction vessel and treated with an aqueous solution of a strong organic acid to effect conversion to the PTH. The PTHs produced from each degradation cycle may be transferred to fraction collector vials until several are manually collected and prepared for analysis. Alternatively, the PTHs may be transferred directly and automatically from the sequencer conversion vessel to an on-line analysis system (Machleidt, W. and Hoffner, H., in Methods in Peptide and Protein Sequence Analysis, pp 35-47, Birr, ed., Elsevier (1980); Wittman-Liebold and Ashman, in Modern Methods in Protein Chemistry, pp 303-327, Tschesche, ed.,de Gruyter (1985); Rodriguez, J. Chromotography 350, pp 217-225, (1985)).

Although a variety of analytical procedures have been used to identify the amino acids released during the Edman degradation, only high performance liquid chromatography (HPLC) is currently in widespread use. In fact, HPLC on reverse phase, silica-based packings has revolutionized peptide sequencing. It provides rapid, sensitive and quantitative analysis of PTH amino acides and is presently the only technique used for PTH analysis that can reliably resolve all of the PTH amino acids in a single chromatograph run. Moreover, because it provides quantitative data at the picomole level, HPLC is the only analytical method suitable for microsequencing by automated Edman sequencers at the present time.

Ideally, each chromatogram would provide a simple qualitative answer, i.e., the presence of one and only one PTH. As a practical matter, this is never the case; each chromatogram contains some amount of all PTHs, and a quantitative evaluation of the relative amounts must be made in order to make the sequence assignment. Several factors give rise to this problem. First, protein or peptide samples are unlikely to be pure. They always contain some level of other peptides or free amino acids that give rise to PTH signals during sequencing. Second, repeated exposure of the sample to the cleavage acid during the Edman chemistry causes splitting of the peptide chain at sites other than the amino terminus. The newly exposed amino terminii resulting from these internal splits then produce PTHs after subsequent coupling/cleavage cycles. As a result, each type of amino acid generally exhibits a background PTH level that slowly changes during the sequence run, typically rising through the early cycles and falling slowly during later ones. The absolute levels of the background are dependent on the amino acid composition of the peptide, the Edman chemistry conditions, and the molecular weight of the peptide. Third, removal of an amino terminal amino acid at any given cycle of the Edman chemistry is incomplete. Therefore, some of the amino acid that should have been released at that cycle will remain for the next coupling/cleavage cycle and be released then. This carryover, or lag, is cumulative; multiple failures on any single peptide molecule will result in a steadily increasing proportion of a population of molecules being out of phase with the expected release order. Fourth, the recovery of the expected PTH is slowly decreased during the run by side reactions that block the amino terminal group, physical loss of peptide from the reaction chamber, internal chain cleavage, and lag. This decrease in signal, measured as the repetitive cycle yield, occurs simultaneously with the increase in noise (due to the factors described above), making correct amino acid assignment ever more difficult as a sequencing run proceeds further into the peptide. Fifth, the relative recoveries of the PTH amino acids from the Edman chemistry vary. Some are recovered almost quantitatively, while others are largely destroyed before analysis.

Despite these problems, rigorous interpretation of the chromatographic data from a sequencer run in terms of an amino acid sequence has not received as much attention as the chemistry and instrumentation employed. Many, perhaps most, sequences are assigned by visual inspection of chromatograms to distinguish the specific increase in the PTH level of one amino acid at each cycle from the general background level of all the PTHs. This method is remarkably simple and effective, but it does have limitations. It relies on the scientist's pattern recognition abilities, skills that are largely subjective and limited to direct comparison of only two to three chromatograms at any one time.

Because of these limitations, an increasing number of scientists are using HPLC peak integration systems to translate the analog signals displayed on chart recorder traces into a simpler set of digital numbers. This allows the recovery of each PTH at each cycle to be plotted on a graph that more clearly shows the specific sequence signals superimposed on the background noise levels. Smithies et al., see Biochemistry 10, 4912, (1971), were the first to define the mathematics of the sequencing chemistry in terms of initial yield, repetitive yield, lag, and amino acid background and to attempt quantitative sequence analysis based on peak integration. Machleidt, W. and Hofner, F., (1981), in High Performance Chromatography in Protein & Peptide Chemistry, pp 245-258, Walter de Gruyter, Berlin, have also contributed to this process, but all of the previous methods have relied on the subjective grading of the integrated peak values by the skilled scientist performing the sequence analysis. The scientist's subjective interpretation of the relative importance of an elevated level of one amino acid versus another at any given cycle has still been required for the final sequence assignment.

In addition to all of the above difficulties having to do with background PTH levels, cumulative lag, side reactions, etc., other important problems are associated with the chromatographic data itself. While most chromatography software available commercially works well with ideal data (i.e. with large, well-resolved peaks), they perform much less well with real world data. With respect to analyses of amino acid derivatives, such non-ideal data is the rule rather than the exception. Generally, amino acid analyses involve separations of a complex mixture of closely-related compounds, frequently at such minute levels that conventional software fails to provide satisfactory results unless the user provides extensive manual input to correct the deficiencies in the software.

In concept, HPLC data systems collect chromatographic data by periodically sampling the output of the HPLC detector and then process this digitized data. Quantitation is then performed using peak integration, which requires locating the start and end points of a peak, measuring the total signal between these points, and subtacting any background signal. The center position of the peak (i.e., its retention time) is also required to identify it as a known component based on retention times obtained with standards. Then, the measured area of a sample peak can be converted to a molar amount based on the measured area of the corresponding standard. This conceptually simple process is, however, complicated by several factors, e.g. such as chromatographic noise, peak overlap, and retention time drift.

The chromatographic noise arises from the detector electronics, incomplete mixing of solvents during gradient chromatography, passage of gas bubbles or particulates through the detector, refractive index changes due to solvent or temperature gradients, and the elution of solvent or column contaminants. At the present time, conventional HPLC systems deal rather imperfectly with both low and high frequency chromatographic noise. Most high frequency filtering relies on hardware implementations and is performed by analog filters built into the detector circuitry, and some HPLC systems attempt to remove low frequency noise (often called baseline drift) by using point-by-point subtraction of a blank chromatogram from the sample data. This latter technique is particularly troublesome since it introduces additional high frequency noise and because baselines can vary substantially from run to run. New and less cumbersome techniques are clearly needed for the reduction of chromatographic noise.

Peak overlap, i.e. incompletely resolved peaks are particularly troublesome to HPLC software. Small peaks that partially overlap larger ones may be missed by the slope threshold routines and hence incorrect chromatogram quantitation can result. When fused peaks are detected, several methods for splitting the total area can be used, which are distinguished by the method by which the baseline under the peak components is set. These include (i) a linear extrapolation between the beginning and end points of the multiplet with a linear drop from the valley between the peaks to the baseline, (ii) a similar extrapolation to set the baseline of a major component with a tangent skim to set the baseline of a minor component, and (iii) linear extrapolations between the beginning and end points of each separate component. The method which gives the most accurate peak measuremnents depends on both the degree of resolution betweeen the peaks and the relative peak heights. It is, therefore, highly sample dependent and frequently requires user adjustment from one sample to another in a set of chromatograms in which these parameters are not constant.

Retention time drift is also a particular problem since once peaks have been located and quantitated, they must be identified by matching their retention times to those of known standards. This is simple if the variation in retention times from run-to-run is always less than the time separation between closely eluting peaks within a run. Typically, software routines are set to search an elution time "window" centered on the standard elution time to find the best match of an unknown peak with the standard. With complex separations that produce closely spaced peaks, this does not always work since elution time drift may move one peak outside its window and place another in it. This problem can be minimized, however, by using easily identified reference peaks to measure the drift and empirically correct the search windows for other peaks. The reference peaks must be well-separated from any neighboring peaks and present in all chromatograms so their search windows can be large enough to allow for the maximum observable drift.

What is needed is a method that resolves most of these problems with chromatogram quantitation and which can be used by a computer to evaluate the set of HPLC data derived from a peptide sequencing run to automatically arrive at an unequivocal call of the sequences, without having to rely on the subjective interpretations of especially skilled individuals.

Summary of the Invention

In accordance with preferred embodiments of the invention, an apparatus and a process are disclosed for unequivocally determining the sequence of a peptide. According to the method, the peptide to be sequenced is degraded cyclicly, arriving at a set of amino acid residues for each cycle. The amount of each amino acid residue is quantitatively measured in each set, then a background level is fit to each cycle to obtain a background fit. A measure of dispersion is then calculated for the background fit, and the measured amounts of amino acid residues in each cycle are normalized relative to the background fit. The largest normalized background-corrected residue amount in each cycle then provides a sequence assignment that can be used for further correction steps if desired. These further steps include correcting at least some of the normalized background-corrected residue amounts for lag into subsequent cycles, thereby obtaining lag-corrected background-corrected residue amounts for each cycle. These lag-corrected background-corrected residue amounts can then be used to correct the original measurements of the amino acid residues in each cycle for differences in injection quantity between cycles, thereby obtaining an injection-corrected residue amount for each cycle. The previous steps of background correction and lag correction are then performed on the injection-corrected residue amounts, and the largest such amount is determined for each cycle to arrive at a definitive sequence assignment for the peptide.

In the preferred mode, the initial step of measuring the amount of amino acid residue in each cycle usually involves several sub-steps. In particular, as the peptide is degraded cyclicly using an Edman degradation scheme, a chromatographic analysis is performed for each cycle, the raw data providing a determination of the amount of amino acid of each kind found at each cycle. In one embodiment, the results of that determination are then subjected to low pass filtering and high pass filtering to remove measurement noise resulting in a baseline-corrected chromatogram. This baseline-corrected chromatogram is then used in the background-correction process.

In another embodiment, the baseline-corrected chromatogram is obtained using a linear, translation-invariant, discrete filter function a_N^α, where N is a measure of the filter width and α determines the resolution enhancement characteristics of the filter function. In that embodiment, the chromatogram is first smoothed with the filter function to suppress high frequency noise thereby providing a second (filtered) chromatogram with a first filtered baseline. Then the second chromatogram is filtered with the parameter α set for peak resolution enhancement to obtain a third chromatogram having substantially the same filtered baseline. The second chromatogram is then subtracted from the third chromatogram, yielding a fourth chromatogram which has the baseline removed. The fourth (baseline-corrected) chromatogram can then be operated on to yield quantitative values of the PTH's to be used in the background-correction process.

The apparatus for carrying out the method of the invention includes a degradation element for degrading the peptide cyclicly to obtain the set of amino acid residues for each cycle, a quantitation element for measuring the amount of each amino acid residue in each set, and a computer element for controlling the degradation element and the quantitation element. The computer element is also for fitting a background level to each cycle, for calculating a measure of dispersion for the background fit relative to the measured amounts of residues in each set, for normalizing the measured amount of each residue relative to the dispersion to obtain normalized background-corrected residue amounts for each cycle, and for identifying the largest normalized background-corrected residue amount in each cycle.

Brief Description of the Drawings

Fig. 1A is a flow chart of a method in accordance with the preferred embodiment of the invention.

Fig. 1B is a diagram of the apparatus of the invention.

Figs. 2A and 2B show the steps of the method for quantitating HPLC peaks.

Figs. 2C and 2D show the steps of the method for quantitating HPLC peaks according to a second embodiment of the invention.

Fig. 3A(1) shows a raw chromatogram of a standard PTH mix.

Fig. 3A(2) shows a baseline-corrected chromatogram performed according to a method of the invention for the standard mix having the raw chromatogram of Fig. 3A(1).

Fig. 3B(1) shows two components of a house filter according to the method of the invention.

Fig. 3B(2) shows the house filter formed by adding the two components of Fig. 3B(1).

Figs. 3C-3I show the results of applying a filtering method according to the invention to the standard PTH mix of Fig. 3A(1).

Figs. 3C′, 3D′, and 3I′ show the results illustrated in Figs. 3C, 3D, and 3I, but on an expanded scale.

Figs. 3J-3M show the results of applying the filtering method according to the invention to a second standard PTH mix.

Fig. 4A and 4B are a flow chart showing the method used for correction of background levels in the amount of each amino acid residue measured for each cycle.

Figs. 5A-5D are a flow chart of the method used to correct for lag of each amino acid residue into subsequent cycles.

Fig. 6 is a flow chart illustrating the method used to correct for differences in injection amounts for different cycles.

Figs. 7A-7C show the results of the method of the invention at different stages of the correction process for the 51st cycle in the degradation of a protein. Fig. 7A shows the injection-corrected raw data. Fig. 7B shows the results of the method after performing background correction on the injection-corrected raw data of Fig. 7A. Fig. 7C shows the results of performing a lag correction after performing the background correction illustrated in Fig. 7B.

Figs. 8A-8C show the results of the method of the invention at different stages of the correction process for glutamic acid at different cycles in the degradation of a protein. Fig. 8A shows the raw data (injection-corrected). Fig. 8B shows the results of the method after performing background correction on the injection-corrected raw data of Fig. 8A. Fig. 8C shows the results of performing a lag correction after performing the background correction illustrated in Fig. 8B.

Detailed Description of the Preferred Embodiments

In accordance with preferred embodiments of the invention, illustrated in Fig. 1A is a flow chart showing a general method for determining the sequence of a peptide chain. In this method, the various steps comprise both direct physical measurements and computational elements based on those measurements. Since the method can be implemented entirely automatically, with the various physical measurements being performed by instruments under computer control, the language of computer programming lends itself to a description of the invention. Hence, to be relatively consistent with that language, hereinafter each step of the method will be referred to either as a "program element" or as a "subroutine", rather than as a "step" as the case may be. The term "subroutine" will be used to describe computer subprograms of the method which are substantially complete in themselves in performing a particular computational task. The term "program element" will be used loosely to describe one or more individual steps of a particular computer program or an identifiable set of steps which make up a complete task, be it a physical measurement or a computation, but which is not in itself necessarily a single subroutine.

The physical apparatus used to implement the method is illustrated in Fig. 1B and includes a sequencing system 51 for performing sequential Edman degradations, a PTH analyzer 53 which provides on-line PTH analysis using HPLC for the samples provided by the sequencing system, and a computer module 55, which acts both as a system controller and as the analysis device for the entire system for performing the various steps used to arrive at a quantitative call for the peptide sequence. As a system controller the computer module is responsible for timing and incrementing degradation events in the sequencing system, for transferring materials to the PTH analyzer, and for controlling the PTH identification process. As an analysis device, the computer module stores the data provided by the PTH analyzer, and performs the various steps described below to eliminate noise and systematic errors from the raw PTH values measured by the PTH analyzer in order to identify the largest relative PTH value in each cycle and thereby determine the peptide sequence. In the preferred mode, the computer module 55 is an Applied Biosystems Model 900A which includes a 16/8-bit microprocessor such as the Intel 8088, a separate math co-processor, 640 kilobytes of RAM, a 10-megabyte hard disk drive, a 360-kilobyte floppy disk drive, a touchscreen CRT, and a graphics printer. Also in the preferred mode, the sequencing system 51 is an Applied Biosystems Model 470A Sequencer and the PTH analyzer 53 is an Applied Biosystems Model 120A.

To begin the method, the peptide sequence to be analyzed is first subjected to an automated Edman degradation cycle at program element 11, and HPLC is performed by the PTH Analyzer at program element 12 to identify the amino acid released. For that Edman cycle, the chromatogram is collected in digitized form by the computer module, and the values of the chromatogram are stored. At program element 13, the cycle number of the degradation just performed is incremented and the process is repeated, until the degradation and chromatograms are performed for all amino acids in the peptide chain, each amino acid corresponding to one degradation cycle. At subroutine 15, each chromatogram is identified and quantitated to determine the amount of each PTH for each cycle of the degradation.

Once the identification and quantitation is complete, a first pass at PTH background noise removal is performed in subroutine 17, and a preliminary sequence assignment is made. This is followed by a lag correction in subroutine 19. The lag correction is made to account for the fact that during the Edman degradation, the removal of the amino acid residue is only partial, so that a fraction of the amino acid appears at subsequent cycles.

Once the lag correction is performed, the remaining PTH values for each cycle can be used to correct for any variation in the amount of sample injected into the PTH analyzer at each cycle, if desired. This injection correction is performed in subroutine 23. Upon completion of the injection correction, the background and lag corrections are repeated and the amino acid assignments are made in program element 25 based on the largest corrected PTH signal at each cycle. The second pass through the background correction and the PTH lag correction can be accomplished in any number of ways. The approach shown in Fig. 1A is to set up a program counter to test at program element 21 to determine if the PTH injection correction has been made, and, if it has been made, to go ahead with sequence selection at program element 25. If it has not been made, the background and lag corrections are repeated. Any of several equivalent program counters can be used to keep track of whether the PTH injection correction has been made. For example, for program element 21, one could use a counter for the background correction subroutine 17, for the lag correction subroutine 19, or for the injection correction subroutine 23 itself. Another less practical, but equivalent approach which is not illustrated is to avoid using a program counter and a program loop at all, and to simply repeat the background correction subroutine 17 and the PTH lag correction subroutine 19 after the injection correction subroutine 23, before making the sequence selection.

Each of the various subroutines and program elements will now be discussed in more detail by referring to the various figures enumerated above in the Brief Description of the Drawings. In addition, further specific details for program elements 17-25 can be found in Appendix I which provides a specific example of a preferred source code for carrying out those program elements.

Quantitation of PTH Yields:

Shown in Figs. 2A and 2B, and in Figs. 2C and 2D are flow charts providing details of a first embodiment and a second embodiment, respectively, of subroutine 15 for identifying and quantitating the HPLC peaks. Although standard HPLC integration routines can be used successfully for identifying and quantitating HPLC peaks when the amount of peptide being sequenced is relatively large (i.e. when the HPLC signal-to-noise levels are high), modern Edman sequencers can work with sample levels that severely test these routines. HPLC peaks can be obscured by both high frequency noise (typically from the UV absorbance detectors used to monitor PTH elution from the analytical columns) and low frequency noise (typically from both temperature-induced detector drift and absorbance/refractive index changes that occur during gradient HPLC elution). Hence, peak analysis should be preceded by chromatogram filtering to minimize the effects of both types of noise.

According to the first embodiment, after each chromatogram is collected in digitized form by the computer module 55, low frequency noise is filtered from it by an adaptation of the method described by Goehner, Anal. Chem 50. 1223 (1978). The entire chromatogram is first piecewise fitted to a polynomial curve of degree N. Then all points that are more than selected amounts (in standard deviation units) above or below the calculated curve are rejected, and a new fit is made using the remaining points. This procedure continues until the standard deviation between actual and calculated points reaches a set minimum level (i.e. the fitted curve has converged on the baseline of the chromatogram). The polynomial coefficients of the calculated curve are then used to calculate a baseline point for each point in the original chromatogram; the difference between the two sets of values represents the baseline-corrected chromatogram.

In practice, the number of original chromatogram points and the number of baseline slope changes (which defines the degree N of the polynomial used for the fit) generally exceeds the practical computing power of microcomputers. Thus, the routine is performed sequentially on overlapping segments of the chromatogram, with each round of the fit routine used to establish the background for a portion of the total chromatogram. Typically, the first 3/9 of the chromatogram points are fit first and used to set the first 5/18 of the baseline points. Then, 3/9 of the points starting at the 2/9 position from the front end of the chromatogram are fit and used to set the next 4/18 of the baseline points. Next, 3/9 of the points starting at the 4/9 position from the chromatogram front are fit and used to set the next 4/18 of the baseline points. Finally, the last 3/9 of the points are fit and used to set the last 5/18 of the baseline points. Since, where possible, only the middle section of each calculated curve is used, discontinuities in the baseline-corrected chromatogram where the individual fitted curves meet are minimal.

This baseline calculation process of the first embodiment is illustrated in detail in Figs. 2A and 2B, which begins at program element 211, where all program counters and constants are initialized. In particular, at least three counters are used, say "i", "k", and "m". A counter i is used to index the particular portion of the chromatogram being fit, a counter k is required to identify the particular Edman cycle being fit, and a counter m is used to keep track of the iteration number of the fit. Similarly, it is necessary to pick the degree N of the polynomial to be used to fit the chromatograms, and to decide on the closeness of fit desired for the background calculation. As a practical matter, N is generally chosen to be about 6, and the closeness of fit is generally chosen to be when the standard deviation of the fitted curve to the background points is less than or equal to a constant K (typically twice the high frequency noise level of the detector output). Once the constants and counters are initialized, the method continues at program element 213 by fitting the i-th portion of a measured chromatogram C (t) corresponding the the k-th Edman cycle. On the first pass, i = k = 1 and a polynomial $P_1^{k i}$ (t,N) (i.e. $P_1^{1 1}$ (t,N) on the first iteration) of degree N is used to fit C_0^k (t) typically using a least squares approach. At program element 215, the standard deviation σ_{i1}^k (C) is calculated for the fit, and a function δ (σ_{i1}^k) is defined which corresponds to the maximum point-by-point deviation allowed. Here $\delta(\sigma_{i1}^k)$ is chosen to be $0.1\sigma_{i1}^k$. The magnitude of the standard deviation σ_{i1}^k is tested at program element 217 to see if it is less than or equal to the chosen constant K. If it is less than K, the counter i is incremented at program element 223. The counter i is then tested to see if it is greater than 4 at program element 225. If it is not greater than 4, the next portion of the chromatogram of the k-th cycle is fit in program element 213, etc.

If at program element 217, the magnitude of the standard deviation is larger than K, the program increments m, the iteration counter, at program element 219. Then at program element 221, a new function $C_m^{k i}$ (t) (in this first pass $C_1^{1 1}$ (t)) is calculated, hereinafter called the reduced chromatogram, by removing all points from the chromatogram for which

$$|C_{m-1}^{k i}(t) - P_m^{k i}(t,N)| > \delta(\sigma_{im}^k).$$

This reduced chromatogram has the same values as the originally measured chromatogram, but its domain is reduced. Then beginning again at program element 213, this reduced chromatogram is fit with a new polynomial, and so forth through the program elements 213 through 225 until the reduced chromatograms for all portions of the chromatogram of the k-th cycle are fit according to the fit criterion established.

Once the polynomial fits are completed for each portion i of the chromatogram of the k-th cycle, the baseline of the k-th cycle is calculated at program element 227, the values of the fitting polynomials P_l^{ki} (t) being used to calculate the baseline points b^k (t) for each point in the measured chromatogram C_o^k (t), where 1 = max.m (the last iteration) for the i-th portion. A background-corrected chromatogram $\underline{C}_o^k$ (t) is then calculated at program element 229 by subtracting the calculated baseline points b^k (t) from the measured chromatogram C_o^k (t), which corresponds to having removed the low-frequency background components from the chromatogram. Those skilled in the art will understand that the above approach for filtering out low frequency noise in this embodiment is but one of many equivalent approaches. For example, any complete set of functions could be used for the fitting function rather than a polynomial.

Once these low frequency components have been stripped from the chromatogram, high frequency noise is removed at program element 231 using a standard fast Fourier transform filter as is known in the art. Then peaks in the filtered chromatogram are detected and quantitated at program element 232. Several approaches can be used. For example, time windows set on the basis of the observed elution times of each PTH in a standard mixture can be used to search the filtered chromatogram $\underline{C}$ (t) to determine the amount of each PTH represented in the chromatogram. The amounts can be determined by standard first derivative peak finding and peak integration procedures known in the art, or by fitting the points in each window to a Gaussian curve and calculating the area under the curve (see Kent et al, in Biotechniques 5, pp 314-321 (1978) or, more simply, the maximum point value in each window can be taken as the peak height for the corresponding PTH. The latter approach is faster but requires good separation of the PTHs by the HPLC system.

Performing the peak location and quantitation yields a transformed chromatogram PTH (k) for each cycle k, which corresponds to the amount of PTH of amino acid of kind j in the cycle k. The program then tests for the cycle number at program element 233; and if all the cycles required to degrade the peptide in the sample are not completed, the cycle number k is incremented at program element 235 and the background-correction process begins again at program element 213 for a new cycle.

Figs. 3A(1) and 3A(2) show the results of using the technique illustrated in Figs. 2A and 2B for this first embodiment to remove low frequency noise. In Fig 3A(1), a raw chromatogram C_o^k (t) is shown with its many peaks as a function of time for a 5-pmol PTH standard. The fitted baseline b^k (t) is shown as a relatively smooth dark solid line at the bottom of the curve C_o^k (t). Fig. 3B(2) shows $\underline{C}_o^k$ (t), the baseline-corrected chromatogram of the sample as determined according to the method of the invention at program element 229.

As an alternative preferred embodiment, another approach can be used to quantitate the chromatogram which utilizes the techniques of digital filtering. Although, digital filtering is generally well known for smoothing and resolution emhancement of noisy spectra, and has been applied specifically to the physical measurments obtained by ENDOR (electron nuclear double resonance), it has not apparently been applied to chromatogram quantitation. (See "Variable Filter for Digital Smoothing and Resolution Enhancement of Noisy Spectra," by Bromba et al, Anal. Chem. (1984), 56,2052-2058, and "Properties of a Variable Digital Filter for Smoothing and Resolution Enhancement", by Biermann et al, Anal. Chem. (1986), 58,536-539, for a general discussion of digital filtering in noise reduction). The particular approach disclosed in these references pertains to the use of digital filtering which is a discrete, linear, translation-invariant convolution operation defined by:

$$Af(k) = \sum_{n=-N}^{N} a(n)f(k-n) \quad ,$$

where A denotes the filter operator, a(n) is the filter function (kernel), f is the unfiltered spectrum, and Af the filtered spectrum. In particular, the filter function is a vertically shifted triangular filter where

$$a(n) = \frac{(2\alpha+1)}{2N+1} - \frac{2\alpha|n|}{N(N+1)} \text{ with } n \leq N \quad .$$

For calculation purposes this filter function is typically separated into two parts

$$a(n) = a1(n) + a2(n)$$

$$\text{where } a1(n) = (N + 1 - |n|)\,\frac{2\alpha}{N(N+1)} \quad ,$$

a triangular filter, and

$$a2\ (n) = -\,\frac{2\alpha(N+1)\ -N}{N(2N+1)} \quad ,$$

which is a rectangular filter. Hereinafter this combination will be called a "house" filter, because filter functions a1(n) and a2(n) when superposed on a graph have the appearance of a house. (See Fig. 3B(1) for an example of a triangular filter superposed on a rectangular filter, using $\alpha = 1$, $N = 9$. Fig 3B(2) shows the resulting house filter.)

In this filter function, N is the filter width and α determines the degree of resolution enhancement. The basis for resolution enhancement with this filter is a reduction in line width, since the frequency response exceeds 1 for $\alpha > 1$ at low frequencies. At higher frequencies the frequency response decreases rapidly and ensures that the high frequency noise in the signal is suppressed. The properties of the filter vary, of course, depending on α and N. Generally, $\alpha = 1/2$ is considered to be particularly well suited for signal to noise enhancement of unknown spectrometric functions and approximates a matched filter and also produces the best high frequency attenuation. Generally, $\alpha = 1$ is the largest value which enables frequency responses not in excess of 1, so that $\alpha = 1$ marks the boundary between smoothing filters and resolution enhancement filters. For $\alpha > 1$ the frequency response of the filter increases beyond the frequency $f = 0$, and then falls off rapidly providing general resolution enhancement. For $\alpha \gg 1$ resolution enhancement increases monotonically with α, but so does noise amplification.

Although these digital filtering techniques could be applied directly to the chromatogram quantitation problem, the result would still contain residual low frequency noise, due to systematic errors inherent in the chromatographic technique at the time of measurement. To avoid that problem, an alternative approach is used which constitutes a second embodiment of the invention. This second approach is illustated in the flow chart of Figs. 2C and 2D.

As for the previous embodiment, the calculation begins by initializing constants and counters this time at program element 238. In particular the counter k corresponding to the Edman cycle number is set to 1, so that the calculation can proceed cycle by cycle. At program element 240, the filter parameter is generally set equal to a number in the interval between 1/2 and 1, and in the preferred mode is set equal to 1. The filter width is set to match the chromatogram peak width N1, so that in the preferred mode, the filter function approximates a Savitzky-Golay filter, corresponding to smoothing without resolution enhancement. At program element 242, the measured chromatogram for cycle k, C_0^k (t), is filtered by convolving it with the house filter kernel. This generates a smoothed chromatogram C_{f1}^k (t), i.e. high frequency noise has been filtered out. Next, α is set equal to a number greater than 1 for resolution enhancement at program element 244, which in the preferred mode in this case has been set equal to 20. At program element 246, the smoothed chromatogram C_{f1}^k (t) is convolved with the new house kernel to yield an enhanced smoothed chromatogram C_{f2}^k (t). This chromatogram typically has substantially the same baseline as the smoothed chromatogram C_{f1}(t) but with enhanced peaks and negative sidelobes, since the house kernel is area preserving. As a result, by subtracting the smoothed chromatogram C_{f1}^k (t) from the enhanced smoothed chromatogram C_{f2}^k (t) at program element 248, one obtains a chromatogram C_{f3}^k (t) which preserves only the peak and sidelobe data. Furthermore, the low frequency baseline noise has been completely removed, thereby eliminating an important source of error in quantitating the chromatograms. Since the peaks are the only information of interest, the sidelobes are chopped off at program element 250, to yield a new chromatogram C_{f4}^k (t). Typically this is done by removing those points below a given threshold (typically zero). At program element 252, the conditions for a new house kernel are established by setting α equal to 1/2 and the width N equal to N1/2, to set up a matched filter for chromatogram C (t). A new chromatogram C_{f5}^k (t) is then calculated using this new kernel at program element 254, which essentially just smoothes the chopped chromatogram C_{f4}^k (t). At program element 256, the peaks of chromatogram C_{f5}^k (t) are detected and quantitated, and at program element 258, these peaks are compared with known reference peaks for different amino acids, in order to correlate each of the peaks in the particular Edman cycle with particular amino acids, thereby obtaining the quantitated values $PTH_{j,o}(k)$. At program element 260, the cycle number K is tested to see if it is greater than or equal to the number of amino acids in the peptide. If it is, the program continues to subroutine 17. If not, the cycle number is incremented at program element 261, and the chromatogram quantitation begins again at program element 240 for the new cycle. The reference peaks used for comparison above are obtained by running chromatograms on pure samples of each amino acid and running the filtering routine for each of them to obtain the filtered reference chromatograms for comparison with the sample chromatograms.

Figs. 3C-3H show the results of applying the above smoothing and baseline subtraction method to the standard PTH mix having the raw chromatogram of Fig. 3A(1). Fig. 3C shows the raw chromatogram of Fig. 3A(1) on a reduced scale. Figure 3D shows the results of filtering the chromatogram of Fig. 3C using a house filter with $N=3$, $\alpha=1$, to remove high frequency noise. Fig. 3E shows the results of applying the house filter again (i.e. applying it to the filtered chromatogram of Fig. 3D), this time with $N=3$, $\alpha=20$. Fig. 3F shows the results of subtracting the chromatogram of Fig. 3D from the chromatogram of Fig. 3E. Fig. 3G shows the chromatogram of Fig. 3F after chopping at zero. Fig. 3H shows the chromatogram of Fig. 3G after smoothing with the house filter with $N=2$, and $\alpha=1/2$. Fig. 3I shows the results of scaling the chromatogram of Fig. 3H (dividing by an arbitrary scale factor of 3) to obtain a peak height more comparable to the peak height of Fig. 3C in order to compare the two curves. To illustrate the sensitivity of the method, these results are shown using an expanded scale in Figs. 3C′, 3D′, and 3I′, which correspond to Figs. 3C, 3D, and 3I, respectively.

To illustrate further the sensitivity of the method using the house filter, the results of applying the method to a 20 pmole PTH standard are shown in Figs. 3J-3M. Fig. 3J shows the raw chromatogram for the sample. Fig. 3K shows the raw chromatogram on an expanded time scale in the time interval between 14.0 and 16.6 minutes, which clearly reveals a substantial high frequency noise content in the measured chromatogram. Fig. 3L shows the results of using the house filter to remove the high frequency noise, with $N=6$, $\alpha=1$. Fig. 3M shows the results of applying the entire digital filtering method to the raw chromatogram of Fig. 3J, and illustrates that the final filtered chromatogram is quite smooth and exhibits peaks that are quite well resolved. The subsequent filtering parameters used were $N=6$, $=20$, for the resolution enhancement, and $N=3$, $=1/2$ for the final smoothing.

PTH Background Correction:

Processing all of the chromatograms from the sequencing run with the routines described above in Figs. 2A and 2B or 2C and 2D produces a data array containing raw values for all of the PTHs at all of the cycles. A plot of any individual PTH versus cycle typically shows a rising and/or falling background level of the PTH with one or more cycles where the PTH value is substantially higher than this background level. This background level can be stripped from the remaining PTH yields by a variation of the recursive least squares fit to a polynomial routine used above for low frequency filtering of the chromatograms. In this variation, the iterations of the fit algorithm are continued until the ratio of the standard deviation between actual and fitted data points for successive iterations is above a set value, say S. Once the iterations are concluded, the calculated background PTH values are subtracted from the raw values to yield background-corrected PTH values.

Next, an estimate of the dispersion of this background corrected data is made by performing three iterations of the least squares polynomial fit routine (with degree of polynomial $=1$). The standard deviation of the last iteration then provides an estimate of the variation in the background level, an estimate that allows assignment of a probability that elevated levels at particular cycles are indeed high by a statistically relevant amount. Once the iterations are concluded, the calculated background-corrected PTH values are divided by the standard deviation of the background fit to obtain normalized background-corrected PTH values. This process is then repeated for each PTH so that the remaining PTH values are also expressed in units of standard deviation above (or below) the values calculated from the fitted background curves. At this point, a preliminary sequence assignment is made for each cycle by picking the PTH whose background-corrected normalized value is statistically highest in that cycle.

This process is illustrated in more detail in the flow chart of Figs. 4A and 4B. Here, the process begins at program element 411, where required constants and program counters are initialized. A program counter, "j", is initialized to the value 1. The counter j is used to index the PTH values to denote an amino acid of kind j. An integer M is set equal to the number of cycles divided by 12 (rounded to the nearest integer), which corresponds to the degree of the polynomial that is used to fit the PTH values. Also, an array $\sigma_{s,-1}$ is set equal to zero for all $s=1$ to 20. In addition, the value of S is set equal to 2, S being the cutoff value of the ratio of the standard deviation between actual and fitted data for successive iterations, in order to determine when the iterations can be terminated in the least squares fit routine. At program element 415, another counter "l" is initialized to zero, the counter l corresponding to the iteration number in the least squares fit routine. At program element 417, $PTH_{j,\ell}$ (k) is fit with a polynomial $Q_{j,\ell+1}(M.k)$ of degree M. Initially this means $PTH_{1,o}(k)$ (the totality of values for amino acid number 1 as a function of cycle) is fit with a polynomial $Q_{1,1}(M.k)$. At program element 419, the standard deviation of the fit for the j-th amino acid on the l-th iteration is calculated and is called $\sigma_{j,\ell}$ (i.e. $\sigma_{1,o}$ initially), and a measure of the fit function Δ $(\sigma_{j,\ell})$ is calculated. Here Δ $(\sigma_{j,\ell})$ is defined to be the standard deviation $\sigma_{j,\ell}$ for points higher than the raw data and twice the standard deviation for points below the raw data. The ratio of $\sigma_{j,\ell-1}$ to $\sigma_{j,\ell}$ is then calculated at program element 421, and, if the ratio is greater than S, the program proceeds to calculate the background corrected normalized PTH amounts at program element 425. If the ratio is less than S, the counter 1 is incremented at program element 422, and at program element 423, a new $PTH_{j,\ell}$ (k) is calculated by removing those points (i.e. cycles) where $|PTH_{j,\ell-1}\ (k)-Q_{j,\ell}\ (k)|> \Delta\ (\sigma_{j,\ell})$. Here $PTH_{j,\ell}\ (k)=PTH_{j,o}\ (k)$ for all points k common to the domains of both functions. Once the points are removed, the fitting routine begins again at program element 417, and the iteration process is continued until the ratio$\sigma_{j,\ell-1}/\sigma_{j,\ell}$ is greater than S. An iteration counter m is then initialized at 424, and, as indicated earlier, at program element 425, the background-corrected raw PTH amounts, $PTH'_{j,o}(k)$, are calculated for each cycle. This is done by subtracting the last fitted value $Q_{j,\ell}(k)$ for the PTH of amino acid j at cycle k, from the measured PTH amount $PTH_{j,o}$ (k). At program element 429, the background-corrected raw PTH amount $PTH'_{j,o}(k)$ is fitted with a first degree polynomial $Z_{j,m+1}(k)(Z_{j,1}(k)$ for the first iteration). The standard deviation of the fit, σ'_{j},m, and a measure of the fit function $\Delta'(\sigma'_{j},m)$ is then calculated at program element 431. Here $\Delta'(\sigma_j', m)$ is typically chosen to be equal to the standard deviation for points falling above the fitted line and to be equal to 1.67 times the standard deviation for points falling below the fitted lilne. The program then tests the value of the iteration counter m at program element 433 in order to stop the iterations at three for this particular sequence of fits. The iteration counter is then incremented at program element 435, and at program element 437 a new $PTH'_{j,m}$ (k) is calculated by removing those points from the background-corrected raw values where the difference between $PTH'_{j,o}$ (k) and the fitted value is greater than the measure of the fit function. This iteration process is then continued twice more, i.e. for a total of three times, and after the third time a normalized background-corrected PTH amount $\underline{PTH}_{j,o}$ (k) is calculated at program element 439 by dividing the background-corrected raw value by the standard deviation calculated on the last iteration. Once the background-corrected normalized PTH amounts are calculated for amino acid j, the counter j is incremented at program element 441 and j is tested to see if it is less than or equal to 20 at program element 442. If j is less than or equal to 20, the program loops back through the background fit and correction process, starting again with program element 415. If j is greater than 20, the method then proceeds to program element 443 where a preliminary sequence assignment is made by finding the maximum normalized PTH amount for each cycle.

Those skilled in the art will understand that there are many ways of performing the background correction. For example, different fitting functions may be used, and a measure of dispersion different from the standard deviation may be used for fitting. The important aspect of this background-correction process is to arrive at a normalized sequence of PTH amounts, so that results from cycle to cycle can be quantitatively compared.

Lag Correction:

After the normalized background-corrected PTH amounts have been calculated and a preliminary sequence assignment has been made, the lag correction is performed. At each cycle, k, of the Edman degradation, the removal of the amino acid residue is only partial, i.e. a fraction of the amino acid appears at subsequent cycles k+1, k+2,...,k+i. At any given cycle, the coupling and/or cleavage failure typically adds 1 to 2% to the out-of-phase signal, or lag. Since these failures are cumulative, the observed lag becomes progressively larger as the sequence proceeds, and in long runs more signal may appear in cycle n+1 than in cycle n. Hence, the lag correction is particularly important for accurate sequence assignments for later cycles.

If one defines "Y_o " as the theoretical initial signal, "b" as the fraction of reaction failure, and "$Y_{1,j}$ " as the yield of amino acid 1 appearing in cycle j, and assumes that no signal is irreversibly lost due to side reactions that prevent the Edman degradation or physical extraction of peptide, then:

$Y_{1,1} = 1Y_o(1-b)$
$Y_{1,2} = 1Y_o\ (1-b)b$

$Y_{1,3} = 1Y_o (1-b)b^2$
$Y_{1,4} = 1Y_o (1-b)b^3$

Similarly, for amino acids 2 and 3:

$Y_{2,2} = 1Y_o (1-b)^2$
$Y_{2,3} = 2Y_o (1-b)^2 b$
$Y_{2,4} = 3Y_o (1-b)^2 b^2$
$Y_{2,5} = 4Y_o (1-b)^2 b^3$
$Y_{3,3} = 1Y_o (1-b)^3$
$Y_{3,4} = 3Y_o (1-b)^3 b$
$Y_{3,5} = 6Y_o (1-b)^3 b^2$
$Y_{3,6} = 10Y_o (1-b)^3 b^3$

If the lag yields are expressed as a ratio to the observed primary cycle yield, these expressions reduce to:

$Y_{1,2}/Y_{1,1} = 1b$
$Y_{1,3}/Y_{1,1} = 1b^2$
$Y_{1,4}/Y_{1,1} = 1b^3$
$Y_{2,3}/Y_{2,2} = 2b$
$Y_{2,4}/Y_{2,2} = 3b^2$
$Y_{2,5}/Y_{2,2} = 4b^3$
$Y_{3,4}/Y_{3,3} = 3b$
$Y_{3,5}/Y_{3,3} = 6b^2$
$Y_{3,6}/Y_{3,3} = 10b^3$

In general terms, if Y_k is the primary cycle yield (i.e. $Y_{k,k}$) and Y_{k+i} is the lag yield (i.e. $Y_{k,k+i}$), then:

$Y_{k+1}/Y_k = ([k+0]/1)b$
$Y_{k+2}/Y_k = ([k+1]/2)([k+0]/1)b^2$

$$Y_{k+3}/Y_k = ([k+2]/3)([k+1]/2)([k+0]/1)b^3$$

$$....$$

$$Y_{k+i}/Y_k = ([k+i-1]/i)...([k+2]/3)([k+1]/2)([k+0]/1)b^i$$

This expression is not strictly correct because of the assumptions that irreversible signal losses are nonexistent and that the failure fraction b is the same at each cycle. However, the former assumption introduces a relatively small error as long as the irreversible losses are less than 10% per cycle and therefore does not interfere significantly with subsequent calculations. The effect of the latter assumption, which is clearly incorrect, is more difficult to evaluate. Empirically, it does not seem to interfere when b is measured at each cycle as a cumulative average lag.

The preferred method of the lag correction is illustrated in Figs. 5A, 5B, and 5C. At program element 511, the preliminary sequence assignment determined from subroutine 17 is used to define the primary cycle yield array Y_k (i.e. Y_k = MAX($PTH_j(k)$)). This preliminary sequence assignment is then used to calculate a working value for cumulative lag, kb, at each cycle. This calculation is set out in program elements 513, 515, and 517. First, at program element 513, $Y_{k,k+1}$ is set equal to $\underline{PTH_j}(k+1)$ for all Edman cycles in the sample, where j is chosen to correspond to the amino acid selected in the preliminary sequence assignment for cycle k. A cycle counter "n" is then initialized at program element 515 so that each cycle is corrected one at a time, and the working values for the lag coefficients are calculated for each remaining cycle (i.e. where $k \geq n$) at program element 517 using the formula $Y_{k,k+1}/Y_k$ = kb(k). In program element 519, these working values of kb(k) are then fitted to a polynomial curve B(k) of degree $Z = N/15$ (rounded to the nearest integer), using the method of least squares. Then the measured values of the lag coefficients that differ from the fitted values by more than one standard deviation are discarded, and the remaining data points are refitted. To accomplish this, the standard deviation, σ_β, of the fitting function B(k) from the actual measurements $Y_{k,k+1}/Y_k$ over the domain N of all Edman cycles is calculated at program element 521. At program element 523, all points k for which the actual value $Y_{k,k+1}/Y_k$ differs from B(k), the fitted value, by more than one standard deviation σ_β, are removed from the domain N, forming a new domain N′. A least squares polynomial fit of kb(k) is then performed at program element 525 using the new domain. At program element 527, the revised fitted lag values B′(k) are generated for all cycles k = 1 to N.

Then the failure fraction b(n), is calculated for cycle n at program element 529, using the fitted values of B′-(k), and at program element 531 b(n) is used to generate the fitted lag amounts G(n,i,b) into the next few cycles using the equation $G(n,i,b) = ([n+i-1]/i)...([n+2]/3)([n+1]/2)([n+0]/1)b^{i}(n)$ or all cycles i until G(n,i,b)<0.01, i.e. until the cycle n+i yield is less than 1% of the cycle n yield. At program element 533, the normalized PTH values are sorted for cycle n to find the three largest values. Y_n^1, Y_n^2, and Y_n^3 and their corresponding lags $Y_{n,n+i}$, Y_n^2, n+i, Y_n^3 $_{,n-i}$. At program element 535, the largest value is tested to see if it is less than three times as large as the next highest value, and if it is not (i.e. it is greater than or equal to 3 times the next highest value), the program leaves the amino acid assignment as it was by setting Y_n equal to Y_n^1 at program element 537. If, however, Y_n^1 is less than 3 times Y_n^2 each of the three largest values Y_n^1, Y_n^2, Y_n^3 are lag corrected in program elements 535 through 557 using the fitted lag based on the original sequence assignment to determine if the lag correction would make any changes in sequence assignment. In particular, each of the fitted lags Y^{Fj}_{n+i} is calculated as $G(n,i,b)Y_n^j$ at program element 543, and if the actual Y_n^j $_{,n+i}$ is greater than zero, the yeild Y_n is corrected for the lag in cycle n+i. This correction is made by adding Y^{Fj}_{n+i} to Y_n^j if $Y_n{}^{j}{}_{,n+i} > Y^{Fj}_{n+i}$ or by adding Y_n^j $_{,n+i}$ to Y_n^j if $Y^{Fj}_{n+i} > Y_n^j$ $_{,n-i}$ at program element 547. This correction process is continued for the next few cycles i past the cycle number n, the criterion for cutoff being that the fitted lag be less than 1% of the actual value as is tested at program element 549. This process continues for each cycle i and each value Y_n^j until each Y_n^j is corrected by replacing Y_n^j with

$$Y_n^{j} + \sum MIN \left\{ \begin{matrix} Y^{f}_{n,n+i} \\ Y^{Fj}_{n+i} \end{matrix} \right. .$$

At program element 553, each of these lag corrections is tested to find the maximum lag corrected value and the sequence value Yn is set equal to that maximum, and the amino acid index is determined in order to select the proper sequence call for that cycle n. Once the sequence call for cycle n is completed, either after program element 537 or after element 553, the fitted lags for that sequence call are calculated at program element 559, for the next few cycles past the cycle n, again using the same termination criterion as before, i.e. until the fitted lag is less than 1% of the actual lag. Cycle n is then corrected for lag from the later cycles n+i that have positive lag values using the same cutoff criteria as before at program element 561 by increasing the cycle n yield by the lesser of the calculated lag amount or the actual cycle yield. Next cycles n+i are corrected for lag from cycle n by replacing the lag value at cycle n+i by the greater of zero or the difference between the observed yield $Y_{n,n+i}$ and the calculated lag amount Y^{F}_{n+i} at program element 563 for all cycles i where the fitted lag coefficient is less than 1%.

At this point, cycle n is fixed and cycle n+1 is free from lag from cycle n that would interfere with the sequence assignment of cycle n+1. At program element 565, the amino acid assignments are corrected based on the calculated lag corrections and at program element 567 the cycle counter n is incremented to calculate the lag for the next cycle. The cycle counter is tested at program element 569 to see if all cycles have been corrected. If they have not, the method returns to program element 517 to determine the empirical lag calculations. The polynomial fit routines 519-525 are repeated based on the new sequence assignments and lag coefficients, and the process is continued until the lag for cycle n+1 is corrected and the lag from cycle n+1 is removed from the next few cycles. This procedure continues until all cycles, one at a time, are corrected for lag. As the lag corrections are made during each pass through the procedure, one more cycle is made free of lag interference with its amino acid assignment, until eventually all cycles can be assigned independent of lag effects. The process then proceeds to the injection correction subroutine 23, or the results of the sequence assignment are output directly depending on whether the injection correction has been made. Those skilled in the art will understand that there are other approximations that can be made in arriving at the effects of the lag correction. For example, instead of using the highest three values of the normalized PTH's, one could use just the highest value, or one could use the two highest values to see if the sequence assignment changes, and if it does, go back and check for other sequence choices. Similarly, one could choose more than the three highest values if after lag correction, the sequence call still appears to be equivocal.

Injection Correction:

Once the background and lag corrections have been made, the remaining PTH values at each cycle can, if desired, be used to correct for any variation in the amount of sample injected onto the PTH analyzer at each cycle. For each cycle, all but the two highest PTH values are averaged. Since the corrected PTH values are in standard deviation units, this average would be near zero if the injection for any given cycle were precise. Any nonzero average for a cycle is used to correct the raw PTH yield data for that cycle by subtracting from each raw PTH value the product of the corrected cycle average and that PTH's standard deviation unit (calculated in the PTH background correction routine). This procedure, in effect, uses the set of nonassigned amino acid values at each cycle as an internal sampling standard. Thus, injection corrections can be made in the absence of any added internal HPLC standard.

Shown in Fig. 6 is a flow chart illustrating the injection correction. First, at program element 611, an array $\{Z_{j,n}\}$ is defined, where the element $Z_{j,n}$ represents the value of the PTH amount of amino acid of kind j at Edman cycle n, as calculated from the lag correction subroutine 19. At program element 613, the array is sorted by amino acid to determine the two largest values, say $Z_{j}{}',_{n}$ and $Z_{j}{}'',_{n}$ for each cycle. The array is then averaged at program element 615, for those amino acids other than the two highest, yielding a column array defined as $\{ \overline{Z}_n \}$. An injection corrected PTH value is then calculated at program element 617 by the equation $INJ_{j,n} = PTH_{j,o}(n) - \overline{Z}_n \sigma_{j,\ell}$, where $PTH_{j,o}$ (n) is the raw PTH value found in subroutine 17 on the last iteration 1 for amino acid j. Finally at program element 619, the array $PTH_{j,o}(n)$ is set equal $INJ_{j,n}$, to set up the name required for the PTH array to be used in subroutine 17.

Once all the injection corrections to the raw PTH data set are made, the PTH background and lag corrections must be recalculated using this adjusted data. Once this is done, the subsequent amino acid assignments should be as error free as is possible given the starting chromatograms.

Utility of the Invention

Appendix II is a sequence of Tables that illustrate at several steps of the method; results of the series of corrections described above. Table 1 shows the raw data resulting from quantitation of 60 cycles of an Edman sequencing of an 18 Kilodalton chain (i.e. through program element 15). Table 2 shows the same raw data with the first column, aspartate, background-corrected according to program element 17. Table 3 shows the results of the next loop through the background correction subroutine 17 in order to correct for background in the second column, asparagine. As described earlier, this background correction process is repeated until the PTH amounts for each amino acid are background-corrected. Then a preliminary sequence assignment can be made. Table 4 shows the results of the background correction.

The circled elements in Table 4 are the maximum PTH values in each cycle and correspond to a first preliminary sequence assignment.

Using the preliminary sequence assignment, the lag correction subroutine 19 is then performed which is followed by the injection correction subroutine 23. Once the injection correction is completed, the injection-corrected sequence of PTH's is then background-corrected at program element 17. The results of the lag correction on the background corrected data for cycles 1-37 are shown in Table 5. For comparison, Table 6 shows the results of the lag correction on the background-corrected data for cycles 1-38. Table 7 shows the results of the lag correction for all 60 cycles. The results of the injection correction for cycle 1 is shown in Table 8. Table 9 shows the results of injection correction for cycles 1 and 2. This injection correction process is then continued for each cycle until all cycles are lag corrected. The results are shown in Table 10. Table 11 shows the results of background correction on the injection corrected data of Fig. 10. Fig. 12 shows the results of lag correction for all cycles of the background corrected data of Fig. 11. Again, the maximum values in each cycle correspond to the sequence assignment, which is seen to be different from the preliminary sequence assignment shown in Table 4, now that the injection correction has been performed.

This difference in sequence assignment can be seen more clearly in Figs. 7A, 7B, and 7C which show results of the above sequence of steps for cycle 51 of the 18 Kilodalton protein. In Fig. 7A, a preliminary assignment would appear to be lysine for that cycle. Once the injection correction has been made, and the background has been corrected, as illustrated in Fig. 7B, the selection for cycle 51 is not so clear, but appears to be histidine. This selection is then confirmed on performing the lag correction as seen in Fig. 7C. The effects of the correction process can also be viewed in terms a particular amino acid, as illustrated in Figs. 8A, 8B, and 8C, which show the results of the correction process for glutamic acid by cycle. Fig. 8A shows the raw data (injection corrected) for glutamic acid by cycle. Fig. 8B shows the results after background correction of the data of Fig. 8A, and Fig. 8C shows the results after both background and lag correction.

Those skilled in the art will appreciate that there are many equivalent ways of implementing the above method and different combinations of apparatus can be used to accomplish the method. For example, many of the computational program elements could be implemented separately from the computer module as long as the apparatus used for those computations were under appropriate control of the computer module. In addition it should be appreciated that the particular program counters and constants chosen in the preferred embodiment may vary, for example depending on the number of cycles being degraded, on the desired accuracy of the calculations, and the desired time to complete the sequence call of the peptide. For these and other reasons, it is intended that the scope of the invention be interpreted with reference to the appended claims and equivalents thereto and not be limited to the specific example chosen to describe it.

Claims

1. A method of obtaining a baseline corrected chromatogram using a discrete, linear, translation invariant filter function a_N^α , where N is a measure of the filter function width and α is a parameter whose value determines signal to noise characteristics of the filter function, comprising the steps of:

performing a chromatographic analysis of a sample to obtain a first chromatogram;

filtering said first chromatogram with the filter function, with N set to approximate the width of peaks obtained in said first chromatogram, and α set to filter out high frequency noise from said first chromatogram to obtain a second chromatogram having a first filtered baseline:

filtering said second chromatogram with the filter function, with N set to approximate the width of peaks obtained in said first chromatogram and α set to resolution enhance peaks in said second chromatogram to obtain a third chromatogram having a baseline which is substantially the same as said first filtered baseline; and

subtracting said second chromatogram from said third chromatogram to obtain a fourth chromatogram which is baseline corrected.

2. The method of claim 1 further comprising:

truncating said fourth chromatogram for values below a preselected threshhold to obtain a fifth chromatogram;

filtering said fifth chromatogram with the filter function with α set to remove high frequency noise and N set equal to the width of peaks in said fifth chromatogram to obtain a sixth chromatogram;

detecting peaks in said sixth chromatogram; and

measuring the height of said peaks in said sixth chromatogram.

3. The method of claim 2 wherein said filter function is a shifted triangular filter function proportional to

$$(N+1-|n|)\ \frac{2\alpha}{N(N+1)} - \frac{2\alpha(N+1)-N}{N(2N+1)} .$$

4. The method of claim 2 wherein peaks detected in said sixth chromatogram are compared with peaks in a reference chromatogram to identify the sample.

5. The method of claim 3 wherein said sample comprises a mixture of amino acid derivatives.

6. The method of claim 4 wherein said filter function is a shifted triangular filter function proportional to

$$(N+1-|n|)\ \frac{2\alpha}{N(N+1)} - \frac{2\alpha(N+1)-N}{N(2N+1)} .$$

7. An apparatus for obtaining a baseline corrected chromatogram of a sample having a plurality of components, comprising:

separation means for separating and eluting said components at different times;

detection means for detecting said components that are eluted and for providing a signal corresponding to retention times in said separation means for said components, said signal hereinafter called a raw chromatogram;

computer means coupled to said detection means for analyzing said raw chromatogram, said computer means comprising:

filter means for filtering said raw chromatogram with a first filter function a_N^α that is a discrete, linear, translation invariant function, where N is a measure of the filter function width and α is a parameter whose value determines signal to noise charachteristics of the filter function;

control means for applying said filter means to said raw chromatogram with N set to approximate the width of peaks obtained in said raw chromatogram, and α set to filter out high frequency noise from said raw chromatogram to obtain a second chromatogram having a first filtered baseline;

said control means also including means for applying said filter means to said second chromatogram with N set to approximate the width of peaks obtained in said first chromatogram and α set to resolution enhance peaks in said second chromatogram to obtian a third chromatogram having a baseline which is substantially the same as said first filtered baseline;

said control means also including means for subtracting said second chromatogram from said third chromatogram to obtain a fourth chromatogram which is baseline corrected.

8. A method for synthesising a peptide in accordance with a predetermined amino acid sequence, characterised in that the sequence has been determined by a method according to any one of claims 1 to 6.

Fig. 1A

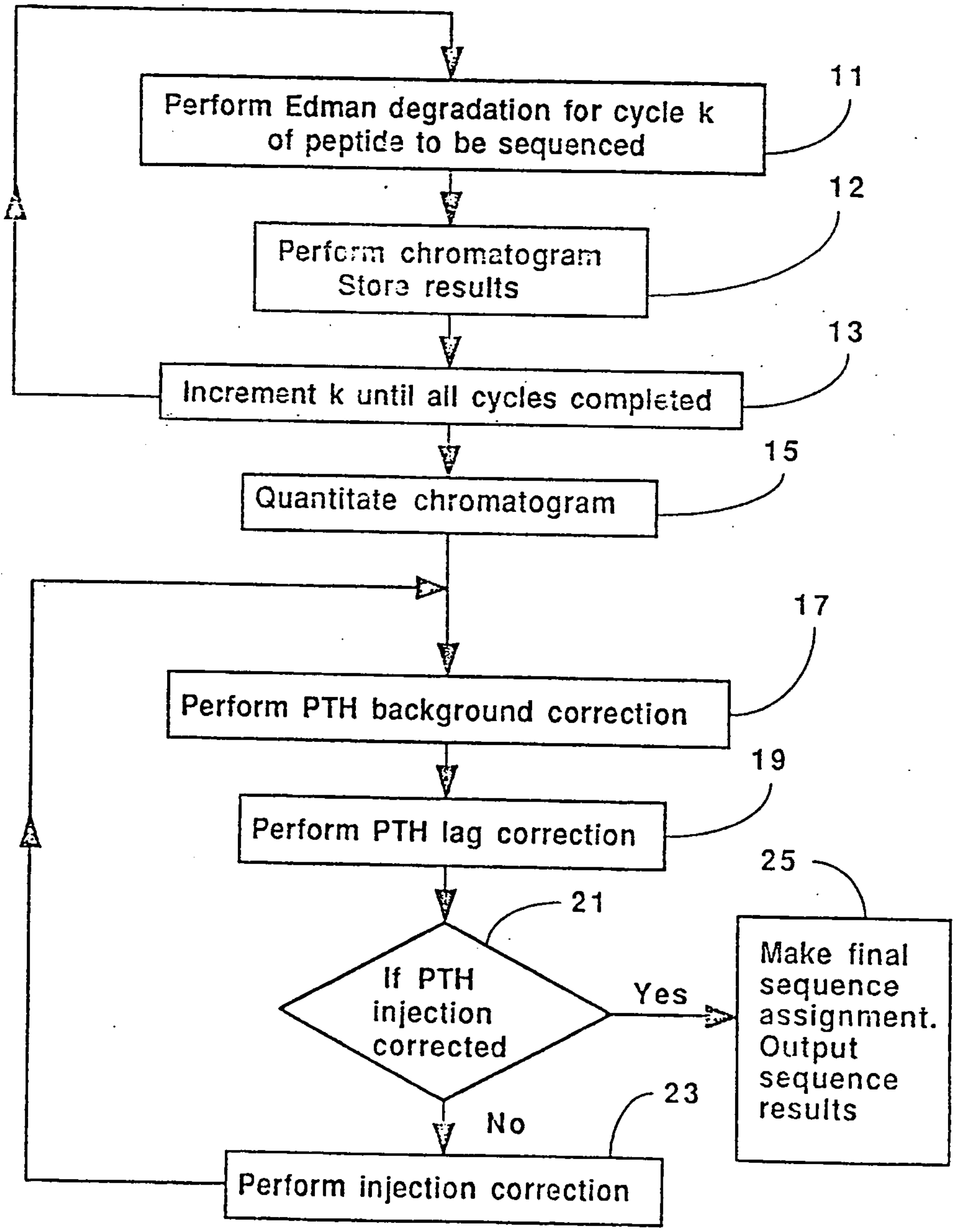

Perform Edman degradation for cycle k of peptide to be sequenced
11
Perform chromatogram Store results
12
Increment k until all cycles completed
13
Quantitate chromatogram
15
Perform PTH background correction
17
Perform PTH lag correction
19
25
21
If PTH injection corrected
Yes
Make final sequence assignment. Output sequence results
No
23
Perform injection correction

Fig. 1B

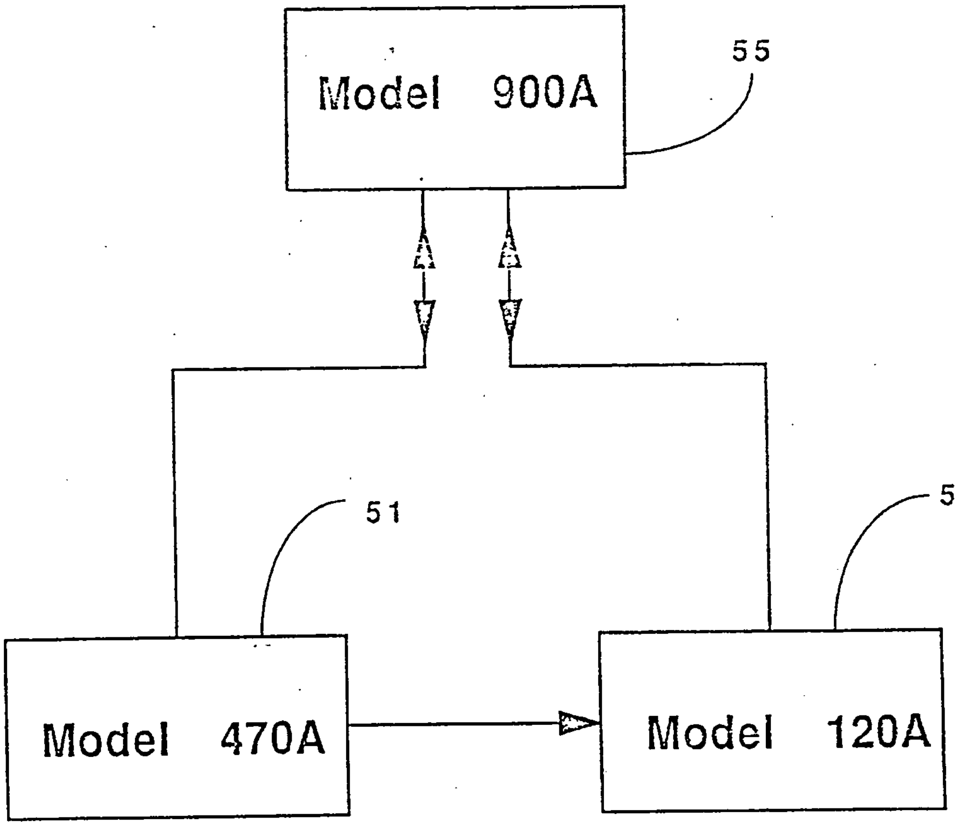


Fig. 2A

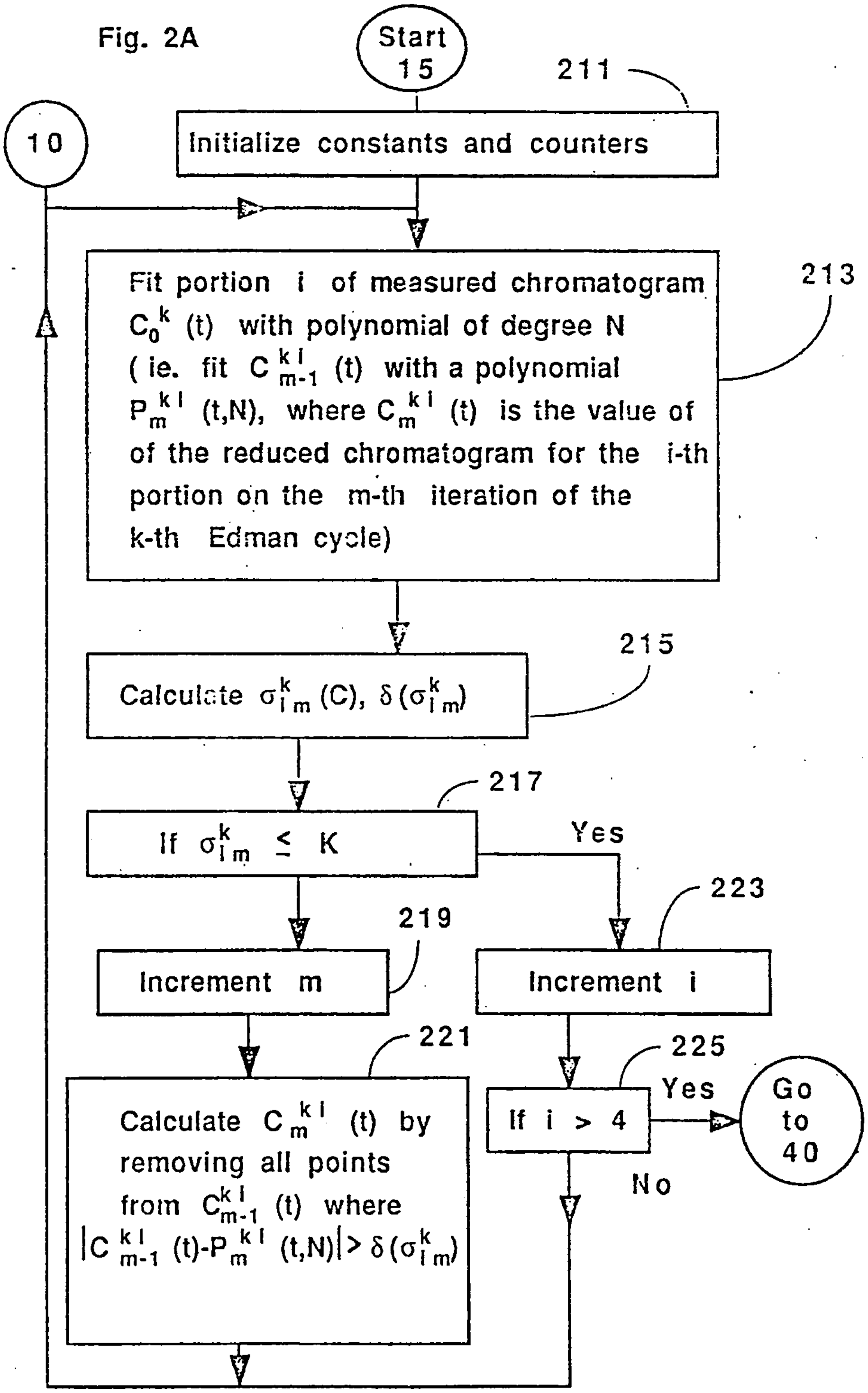
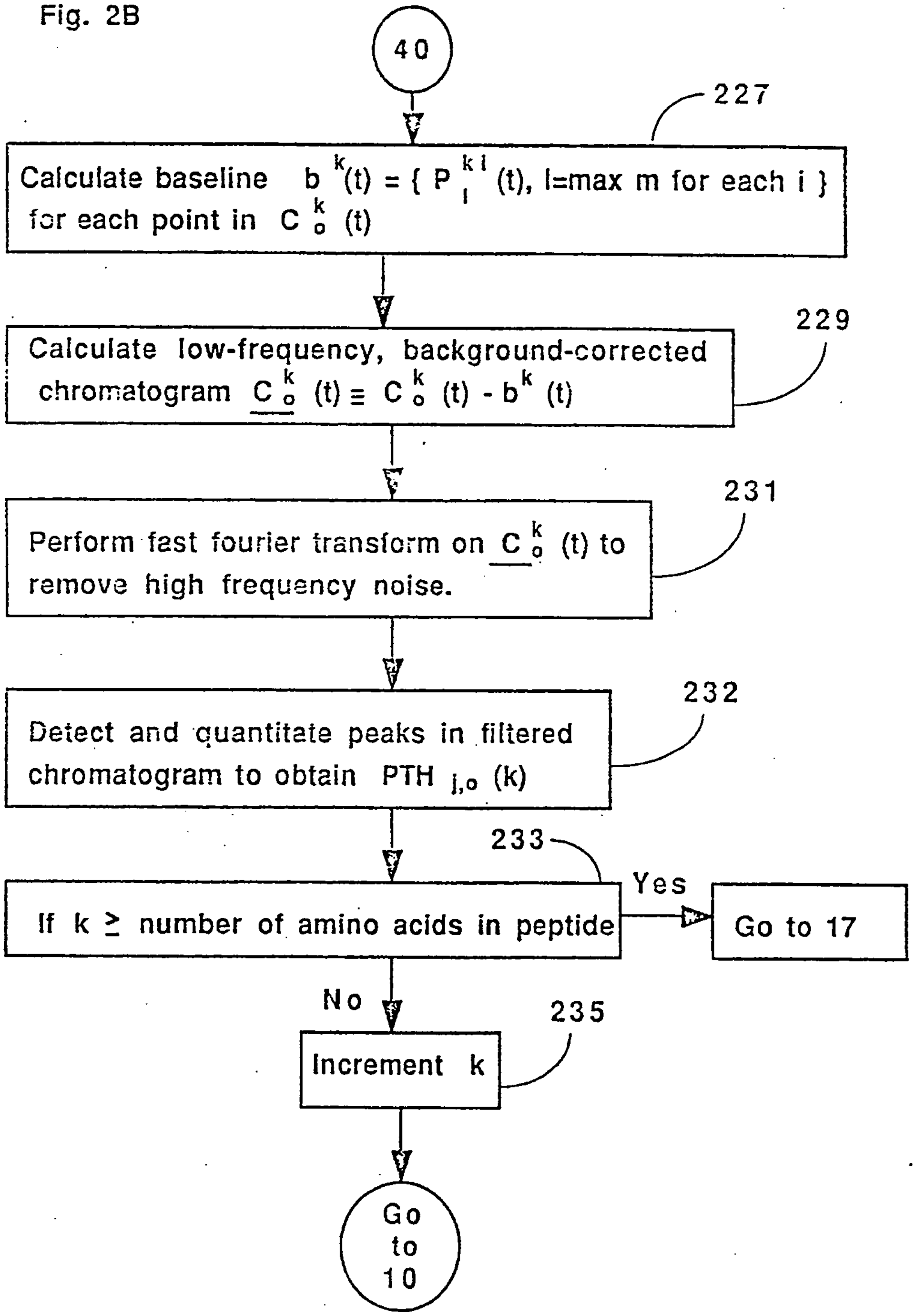

Start 15
211
Initialize constants and counters
10
213
Fit portion i of measured chromatogram C_0^k (t) with polynomial of degree N (ie. fit $C_{m-1}^{k\,l}$ (t) with a polynomial $P_m^{k\,l}$ (t,N), where $C_m^{k\,l}$ (t) is the value of of the reduced chromatogram for the i-th portion on the m-th iteration of the k-th Edman cycle)
215
Calculate $\sigma_{l\,m}^k$ (C), $\delta(\sigma_{l\,m}^k)$
217
If $\sigma_{l\,m}^k \leq K$
Yes
219
Increment m
223
Increment i
221
Calculate $C_m^{k\,l}$ (t) by removing all points from $C_{m-1}^{k\,l}$ (t) where $|C_{m-1}^{k\,l}(t)-P_m^{k\,l}(t,N)| > \delta(\sigma_{l\,m}^k)$
225
If i > 4
Yes
Go to 40
No

Fig. 2B


40
227
Calculate baseline $b^{k}(t) = \{ P_{l}^{k\,i}(t)$, l=max m for each i $\}$ for each point in $C_{o}^{k}(t)$
229
Calculate low-frequency, background-corrected chromatogram $\underline{C}_{o}^{k}(t) \equiv C_{o}^{k}(t) - b^{k}(t)$
231
Perform fast fourier transform on $\underline{C}_{o}^{k}(t)$ to remove high frequency noise.
232
Detect and quantitate peaks in filtered chromatogram to obtain $PTH_{j,o}(k)$
233
If $k \geq$ number of amino acids in peptide
Yes
Go to 17
No
235
Increment k
Go to 10

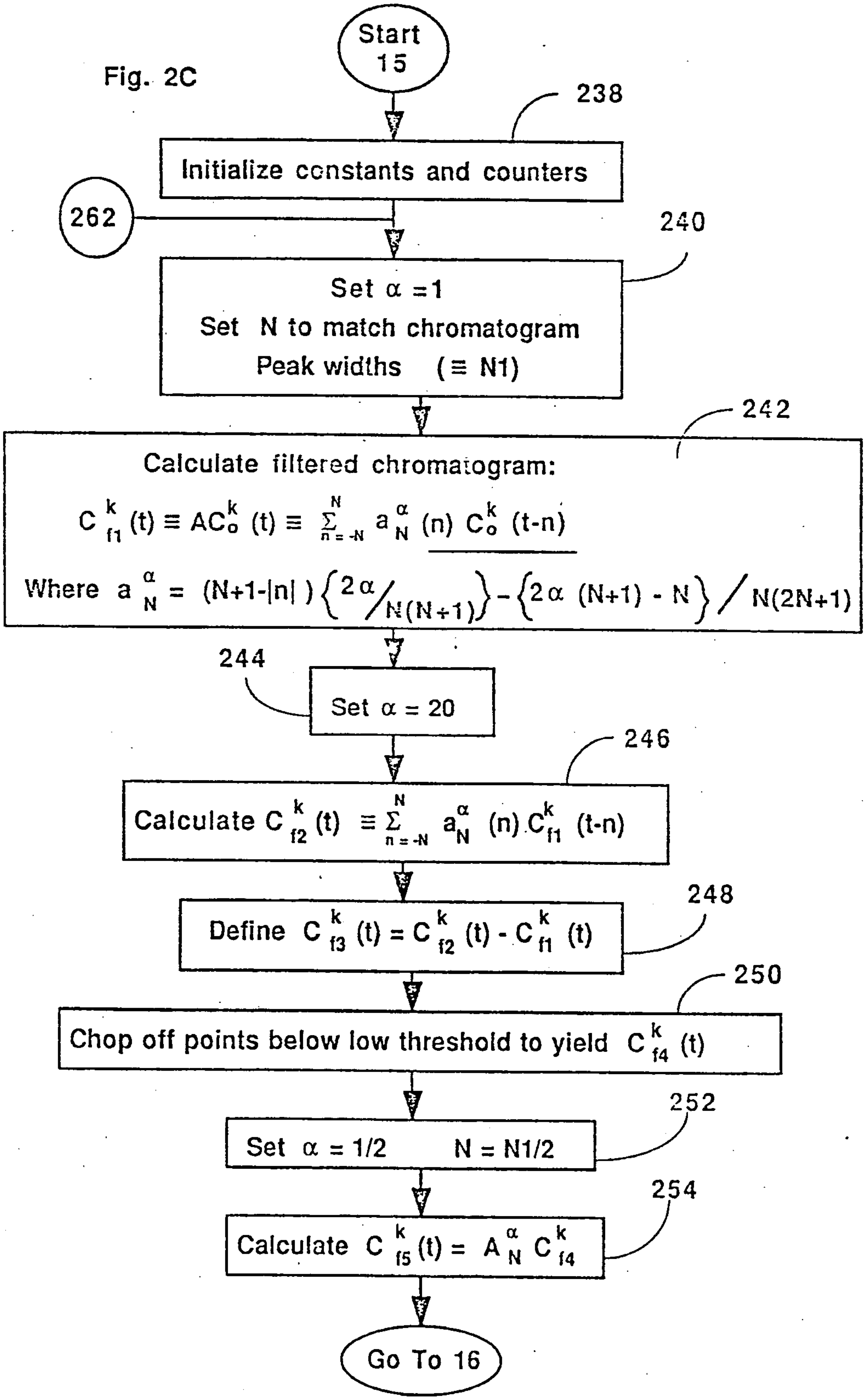
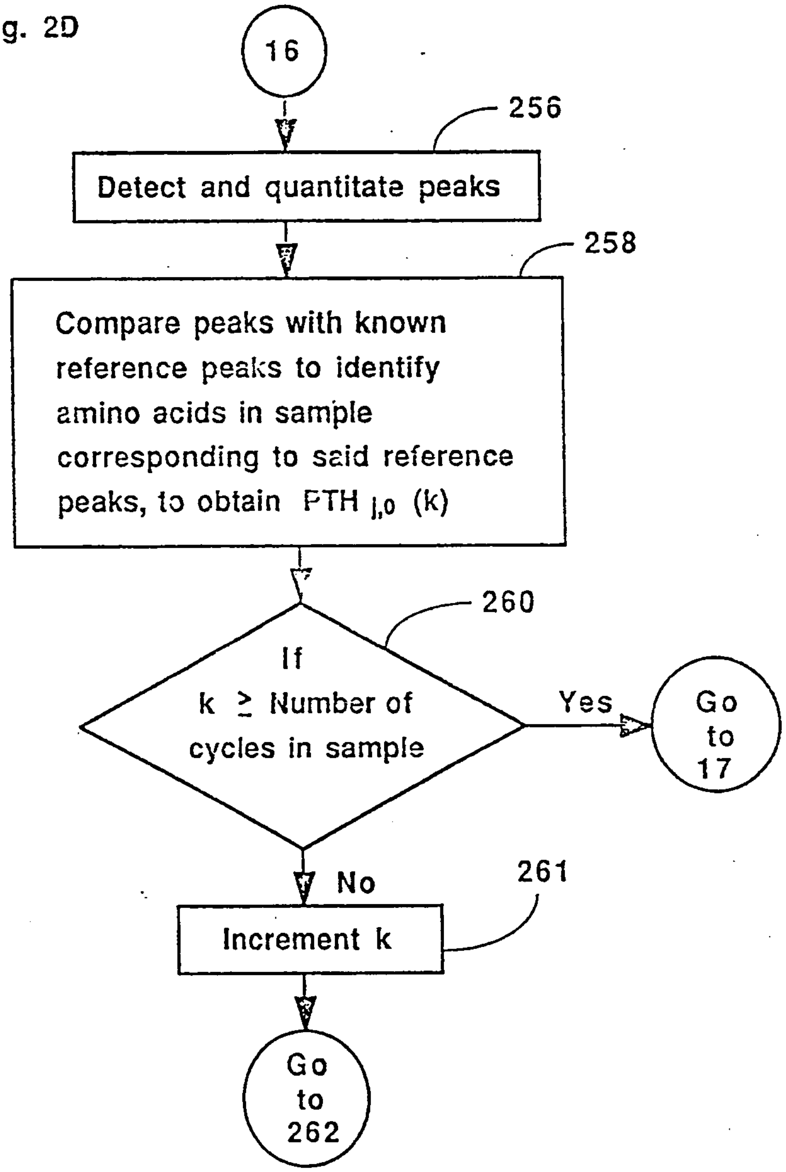


Fig. 2C

Start 15
238
Initialize constants and counters
262
240
Set $\alpha = 1$
Set N to match chromatogram
Peak widths $(\equiv N1)$
242
Calculate filtered chromatogram:
$C^{k}_{f1}(t) \equiv AC^{k}_{o}(t) \equiv \sum_{n=-N}^{N} a^{\alpha}_{N}(n)\, C^{k}_{o}(t-n)$
Where $a^{\alpha}_{N} = (N+1-|n|)\left\{2\alpha / N(N+1)\right\} - \left\{2\alpha\,(N+1) - N\right\} / N(2N+1)$
244
Set $\alpha = 20$
246
Calculate $C^{k}_{f2}(t) \equiv \sum_{n=-N}^{N} a^{\alpha}_{N}(n)\, C^{k}_{f1}(t-n)$
248
Define $C^{k}_{f3}(t) = C^{k}_{f2}(t) - C^{k}_{f1}(t)$
250
Chop off points below low threshold to yield $C^{k}_{f4}(t)$
252
Set $\alpha = 1/2$ $N = N1/2$
254
Calculate $C^{k}_{f5}(t) = A^{\alpha}_{N} C^{k}_{f4}$
Go To 16

Fig. 2D


16
256
Detect and quantitate peaks
258
Compare peaks with known reference peaks to identify amino acids in sample corresponding to said reference peaks, to obtain PTH j,0 (k)
260
If k ≥ Number of cycles in sample
Yes
Go to 17
No
261
Increment k
Go to 262

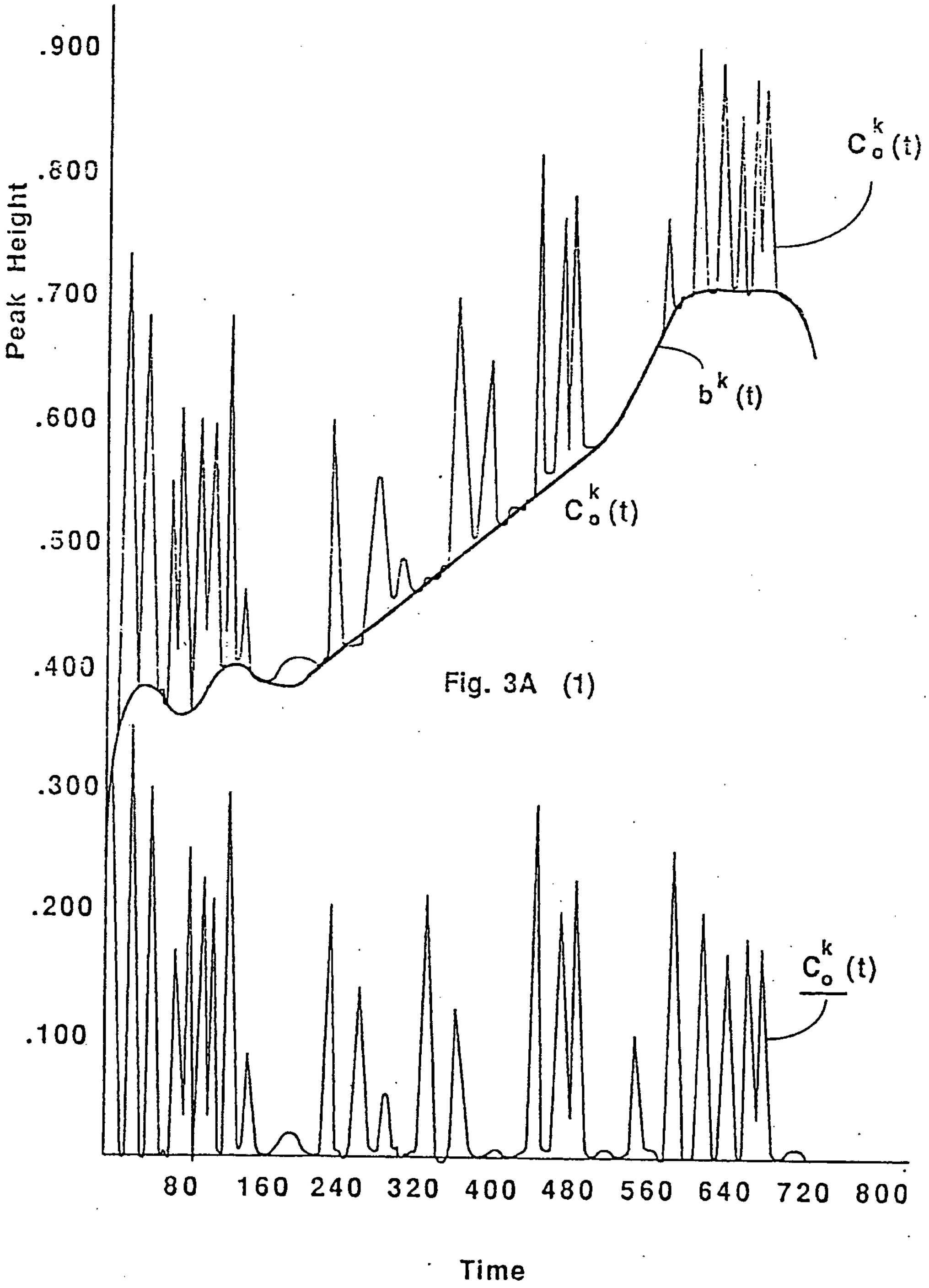


Fig. 3A (1)

Fig. 3A (2)

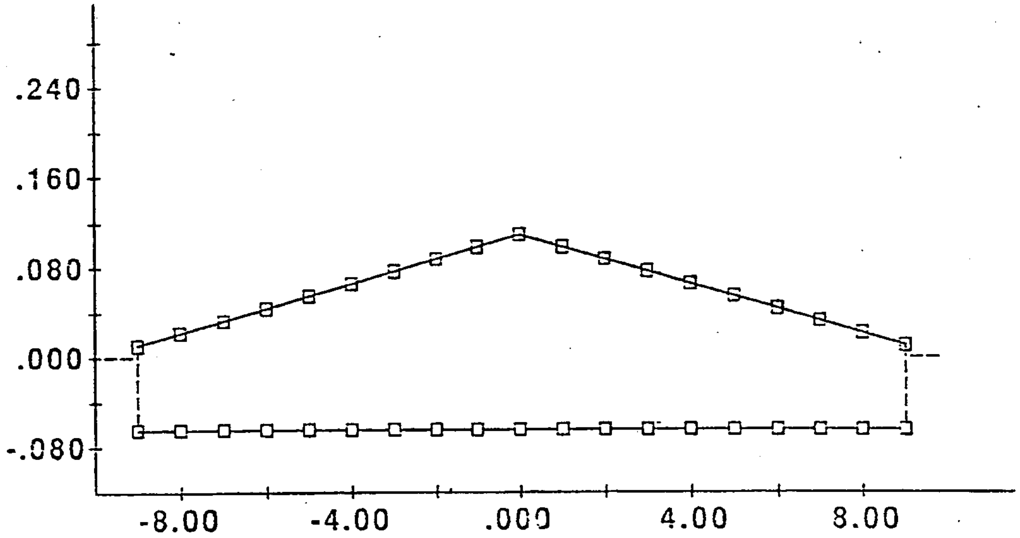


Fig. 3B (1)

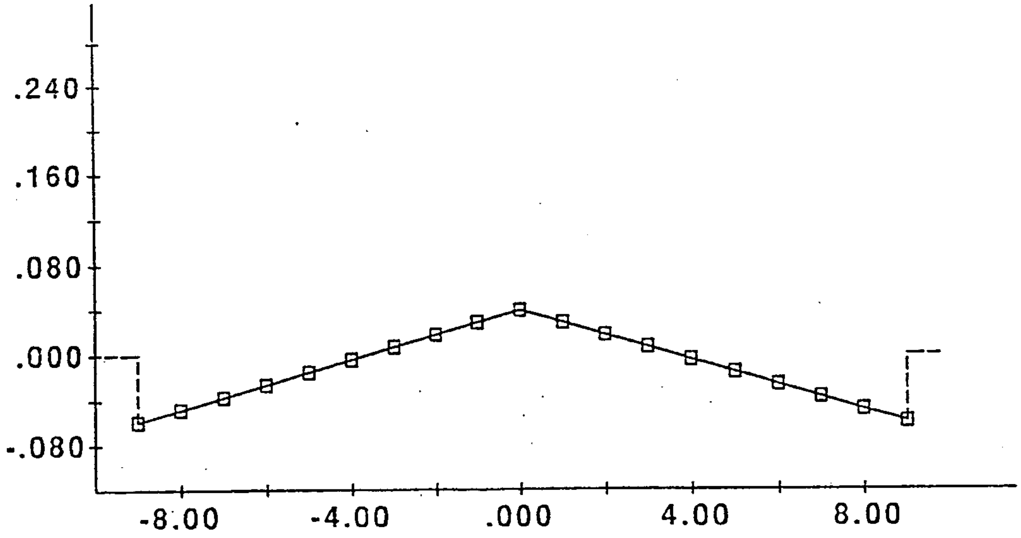


Fig. 3B (2)

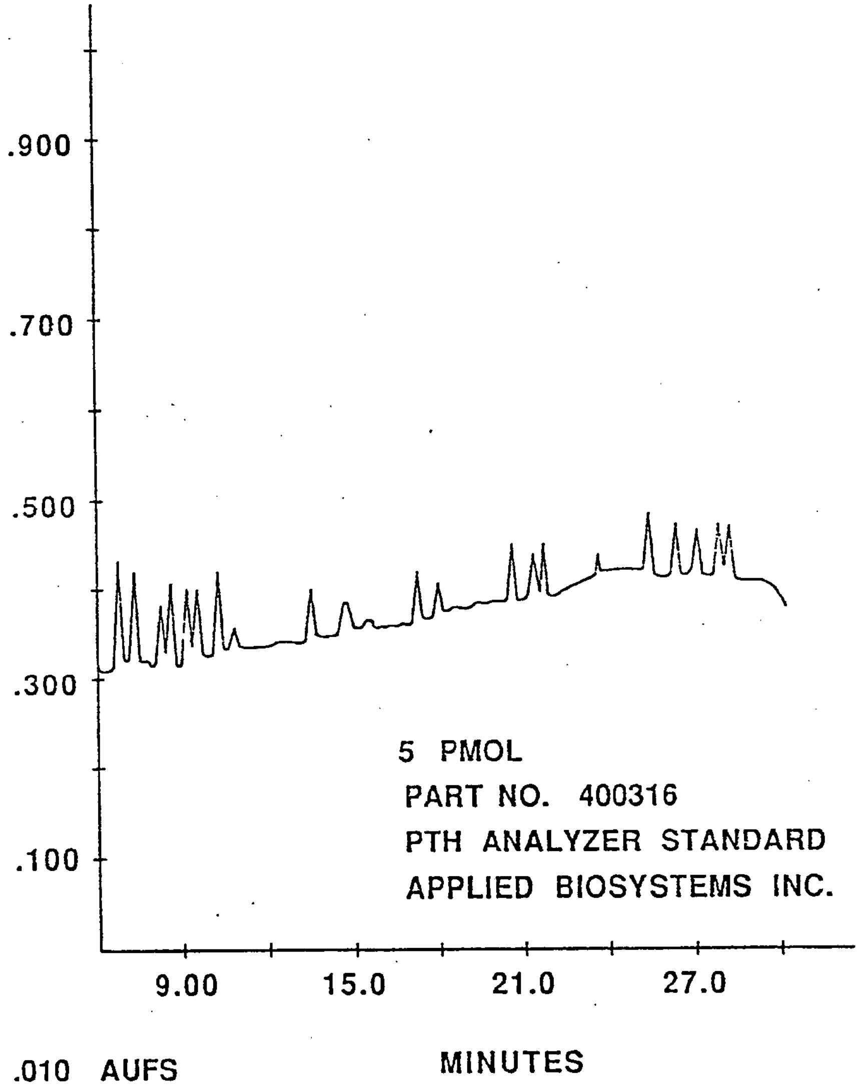

.900
.700
.500
.300
.100
5 PMOL
PART NO. 400316
PTH ANALYZER STANDARD
APPLIED BIOSYSTEMS INC.
9.00
15.0
21.0
27.0
.010 AUFS
MINUTES

Fig. 3C

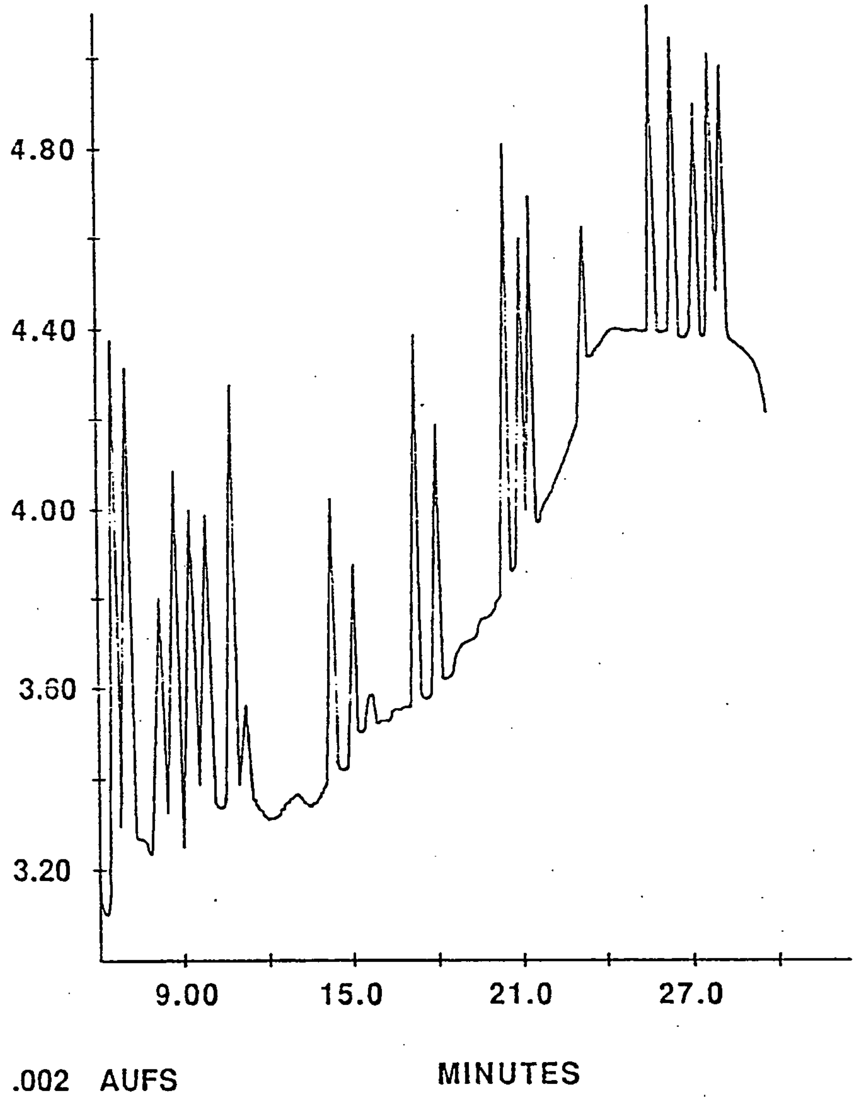


Fig. 3C'

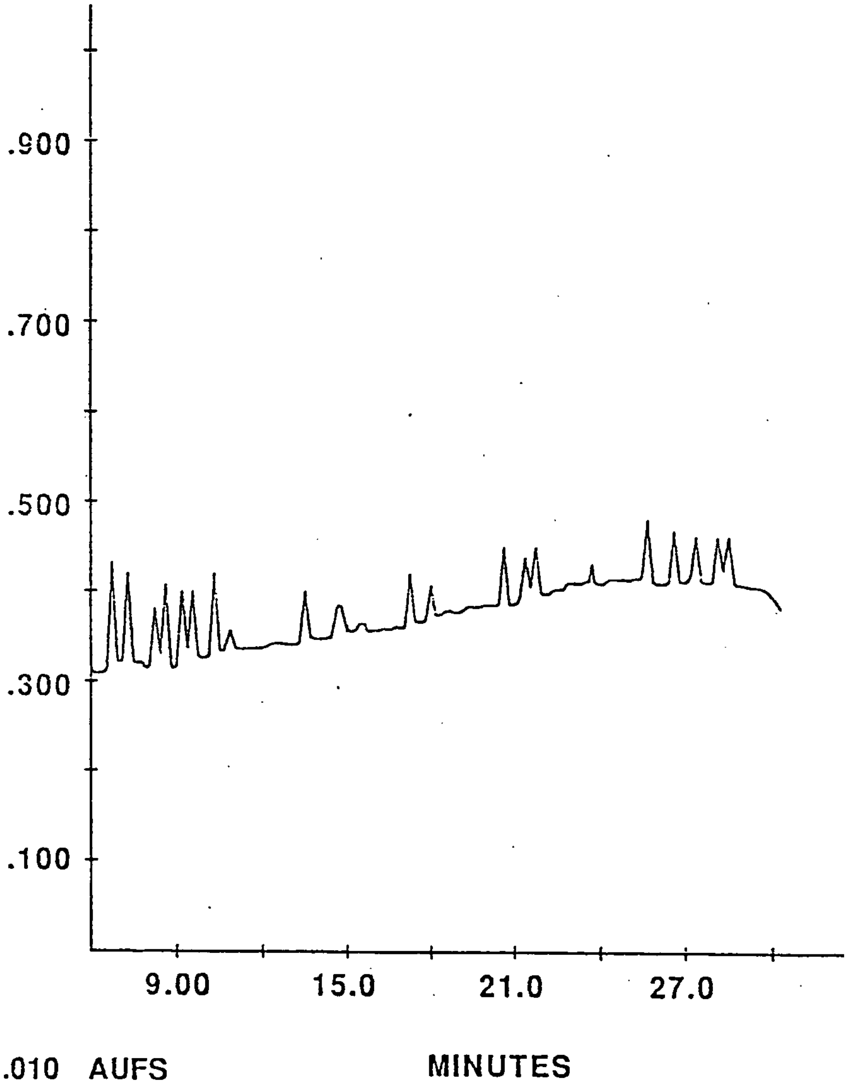


Fig. 3D

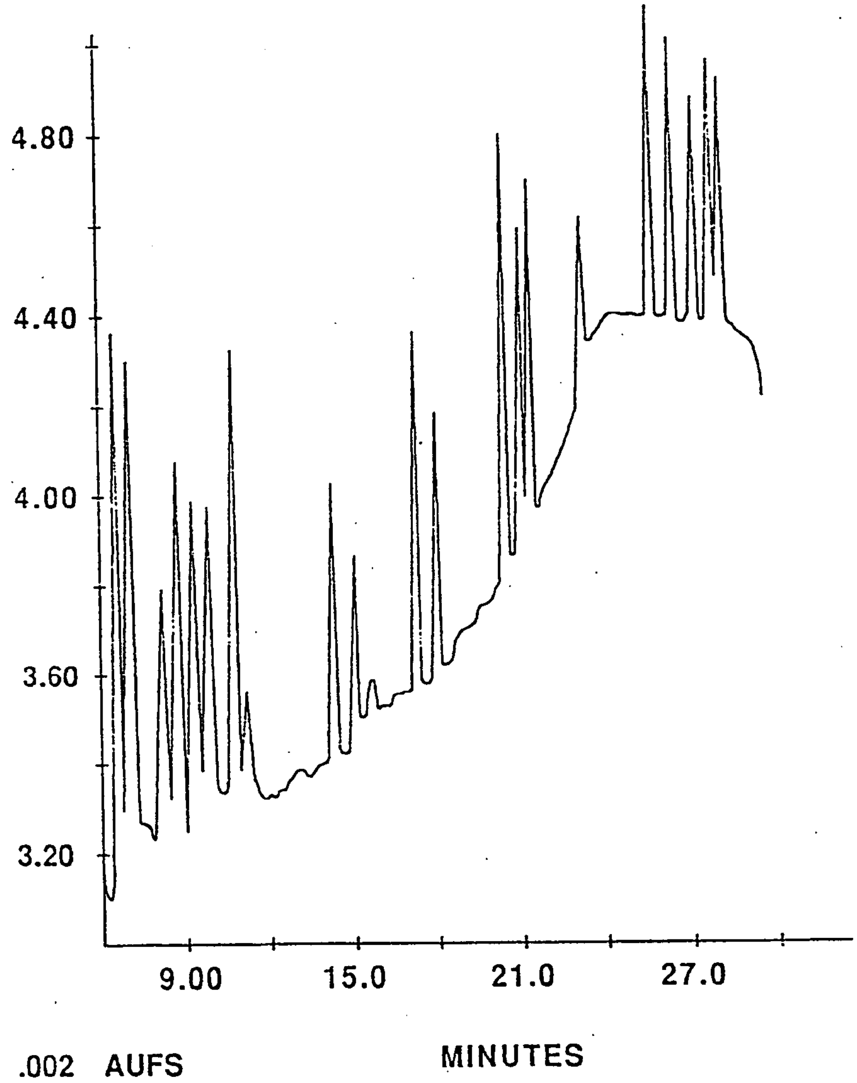
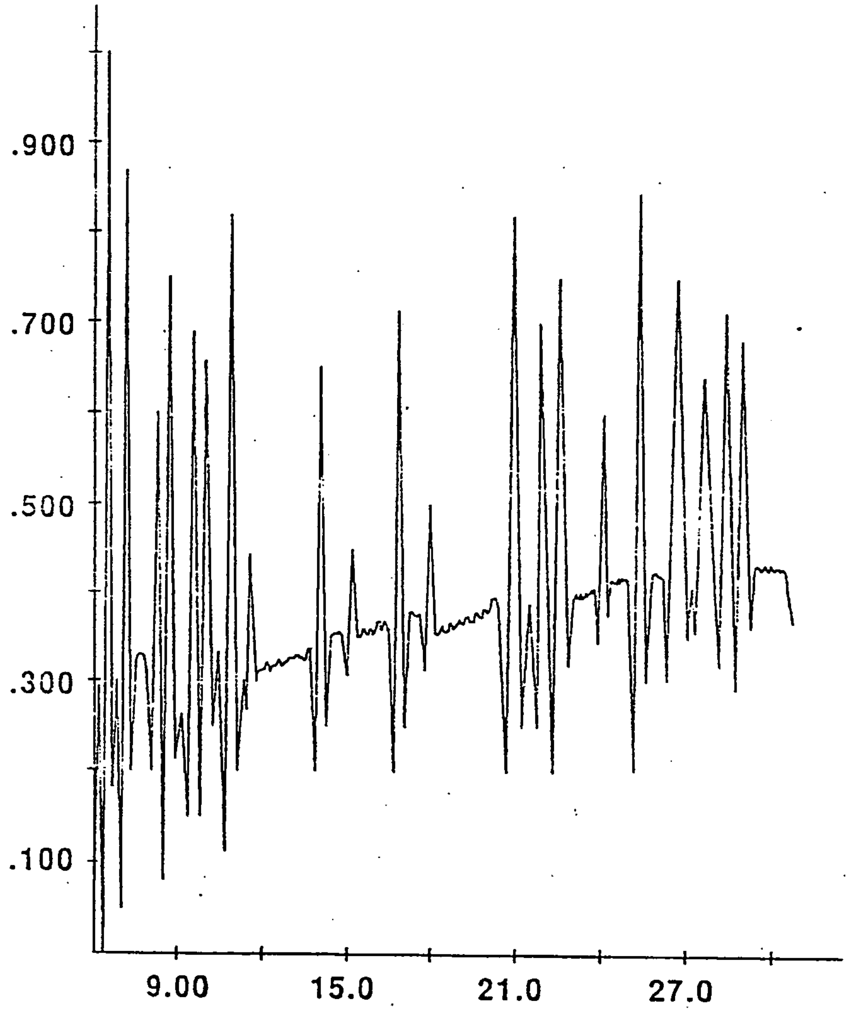


Fig. 3D'



.010 AUFS MINUTES

Fig. 3E

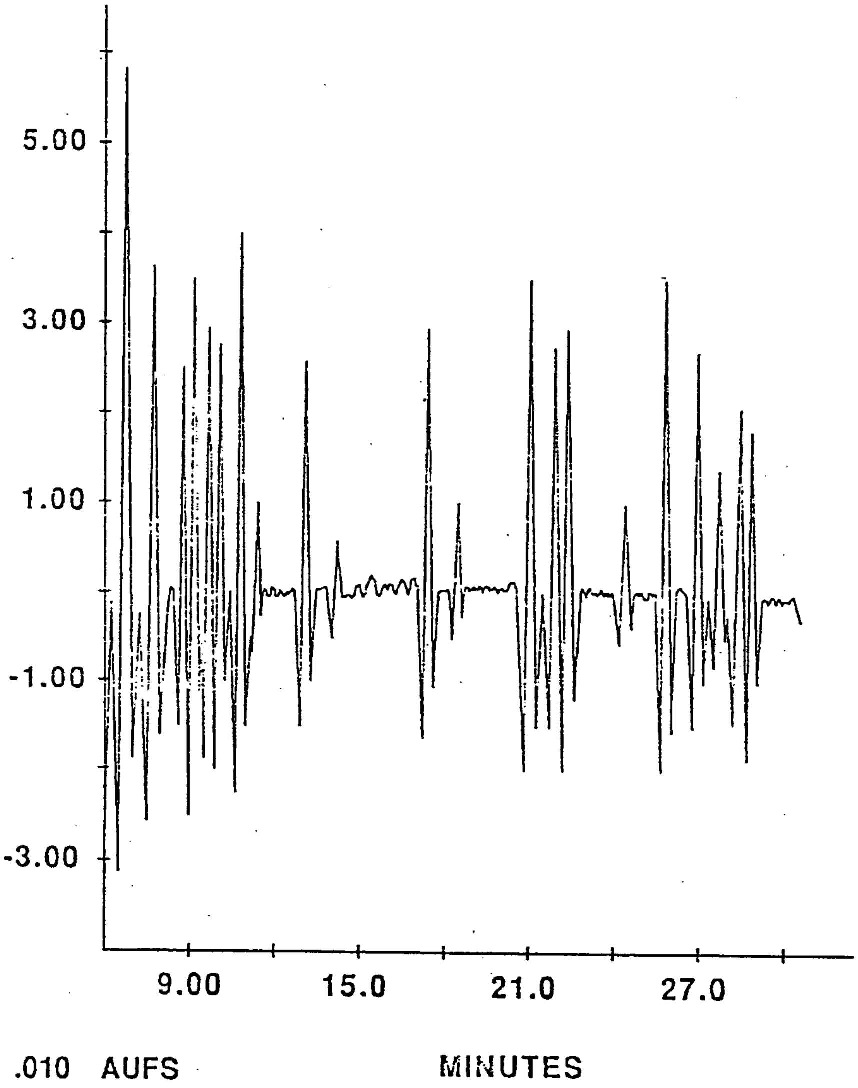


Fig. 3F

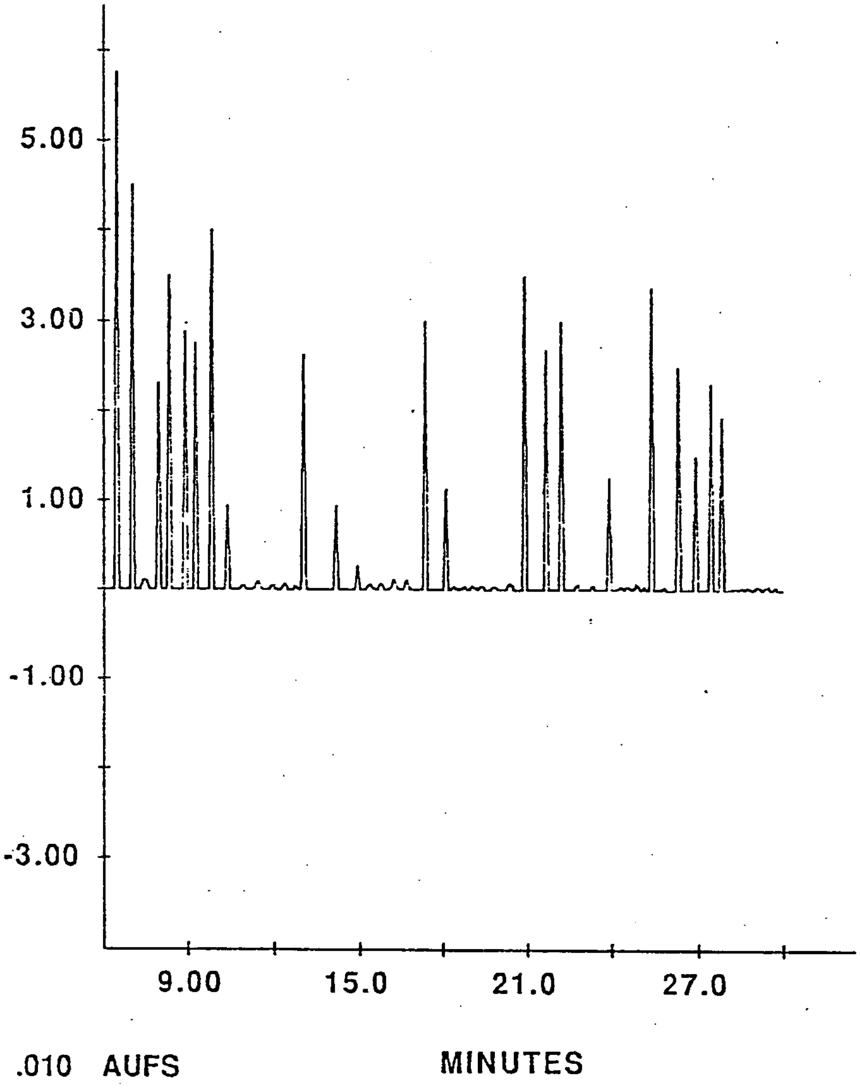


Fig. 3G

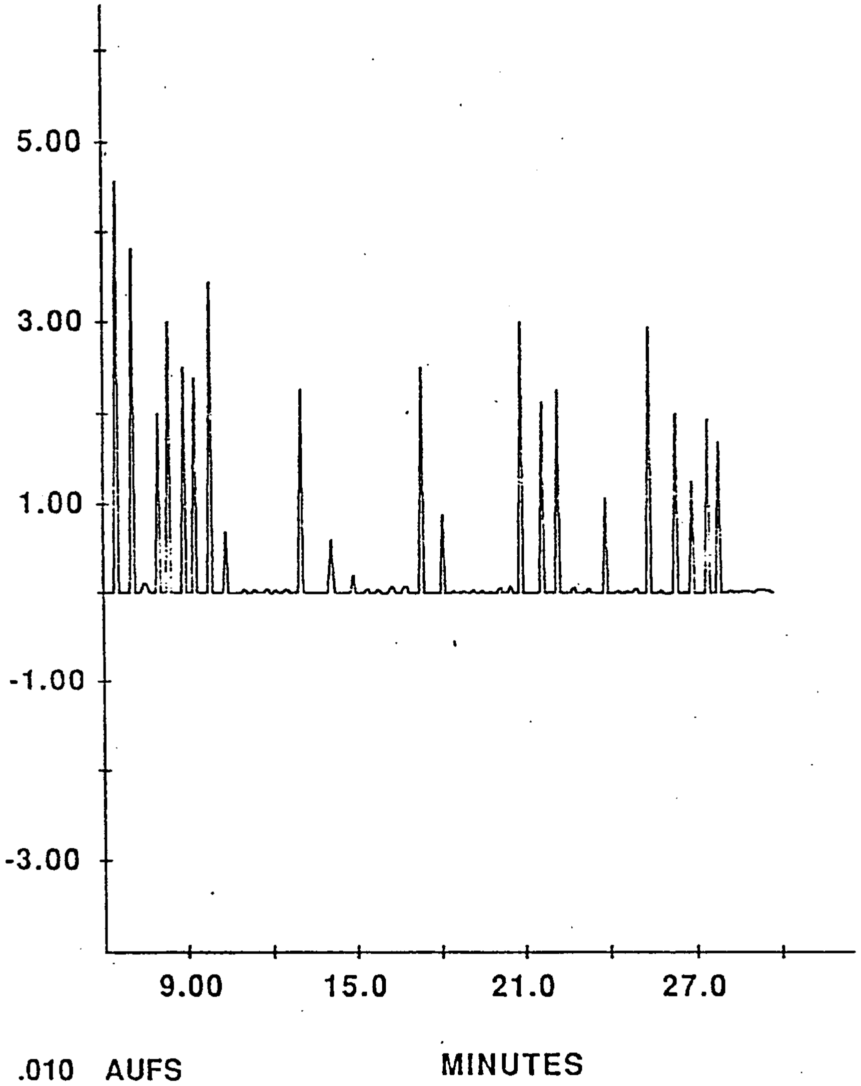


Fig. 3H

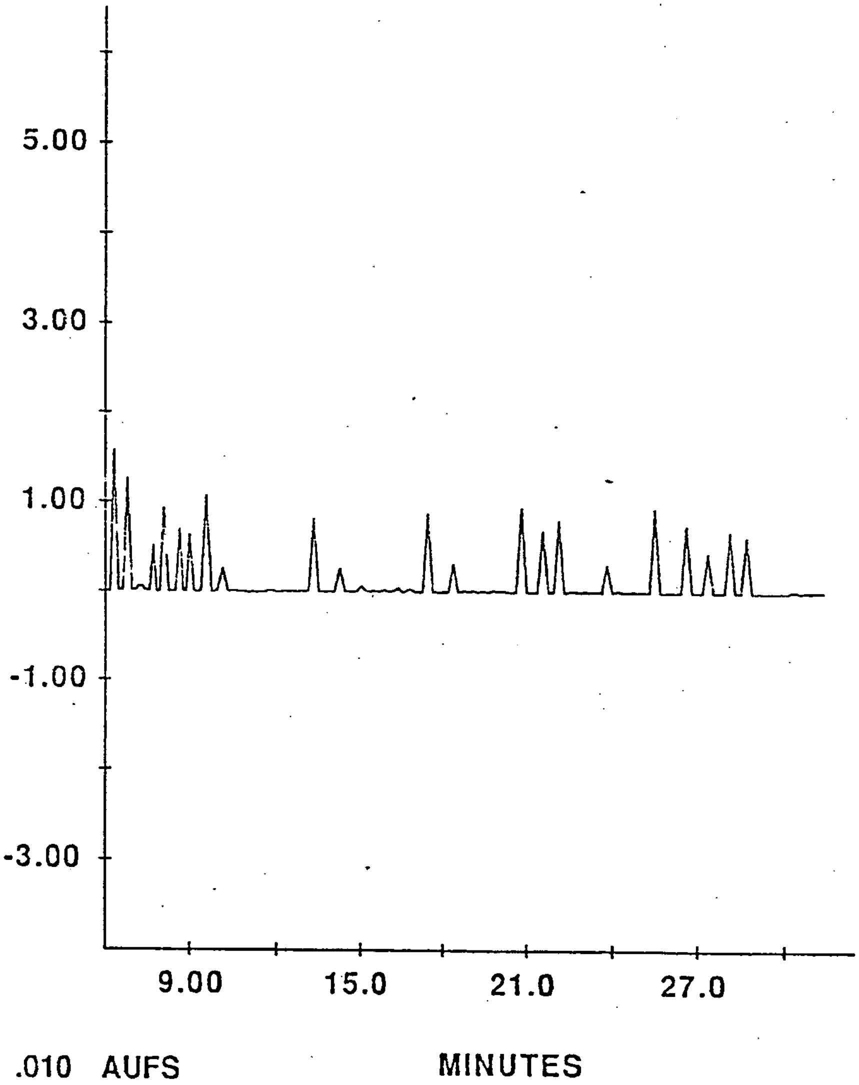


Fig. 3I

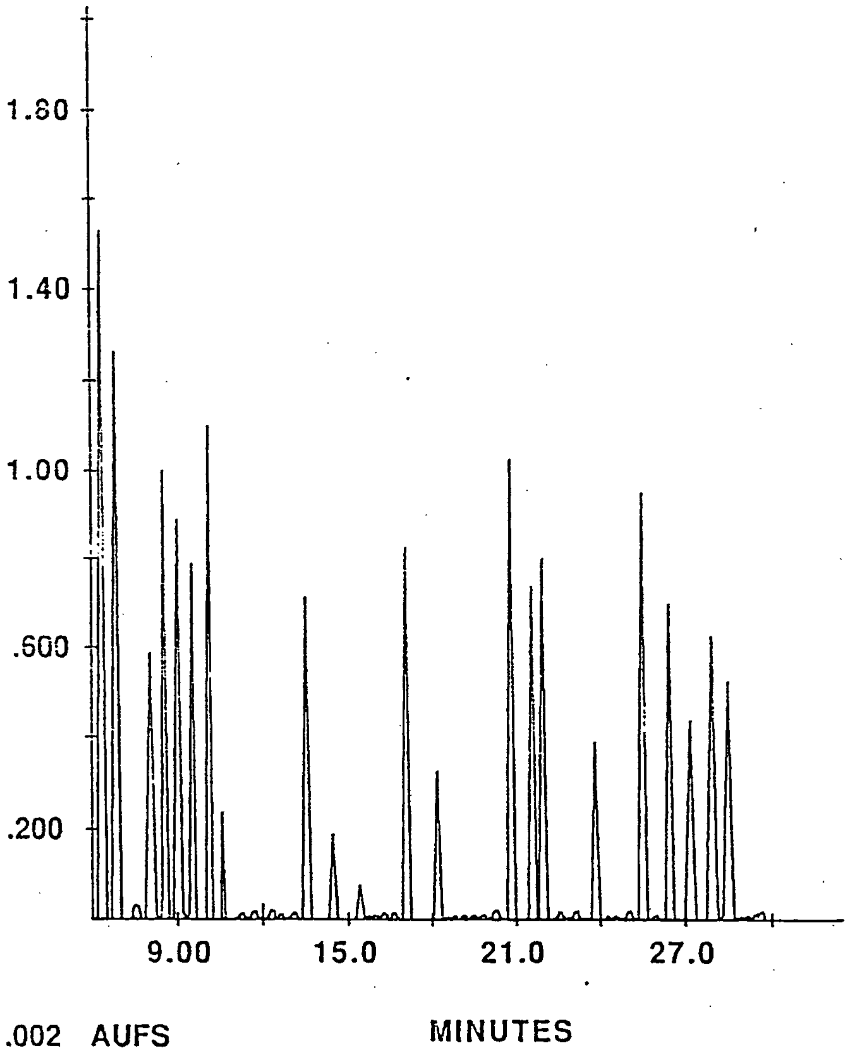


Fig. 3I'

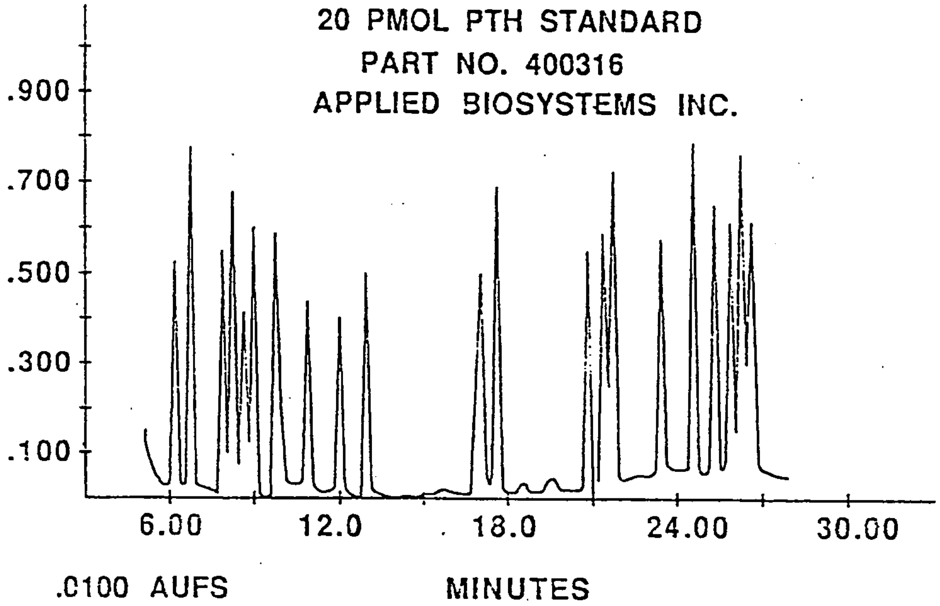

20 PMOL PTH STANDARD
PART NO. 400316
APPLIED BIOSYSTEMS INC.
.900
.700
.500
.300
.100
6.00
12.0
18.0
24.00
30.00
.0100 AUFS
MINUTES

Fig. 3J

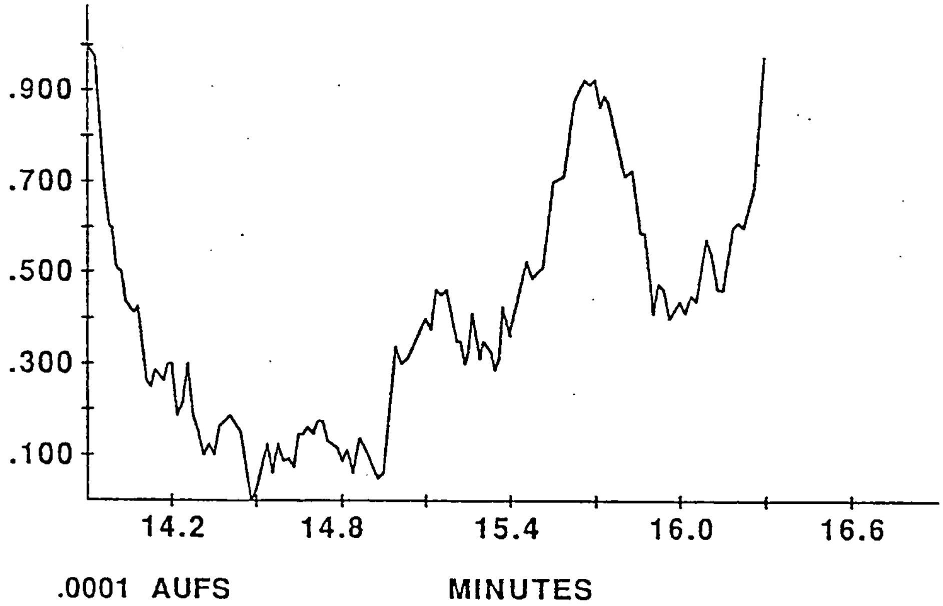

.900
.700
.500
.300
.100
14.2
14.8
15.4
16.0
16.6
.0001 AUFS
MINUTES

Fig. 3K

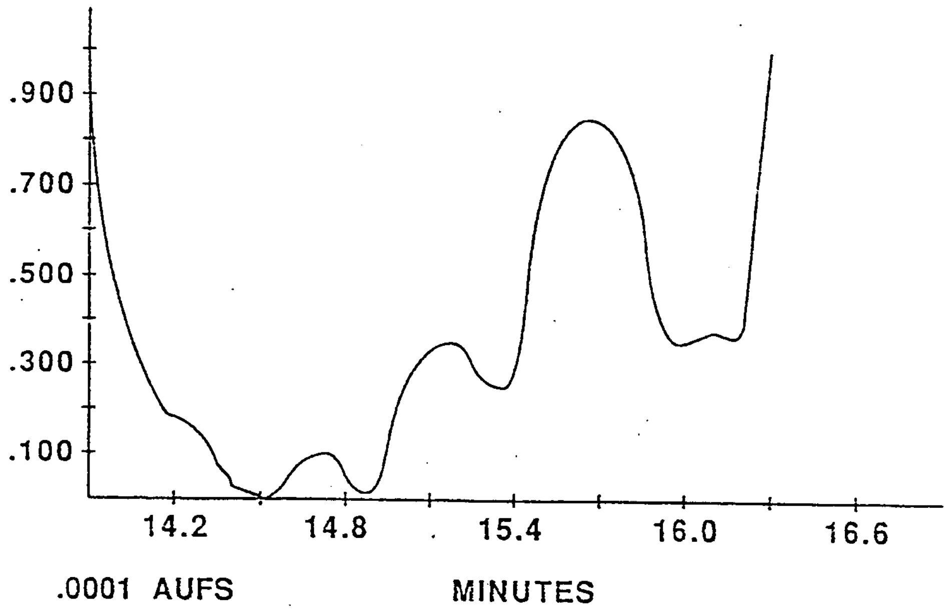

.900
.700
.500
.300
.100
14.2
14.8
15.4
16.0
16.6
.0001 AUFS
MINUTES

Fig. 3L

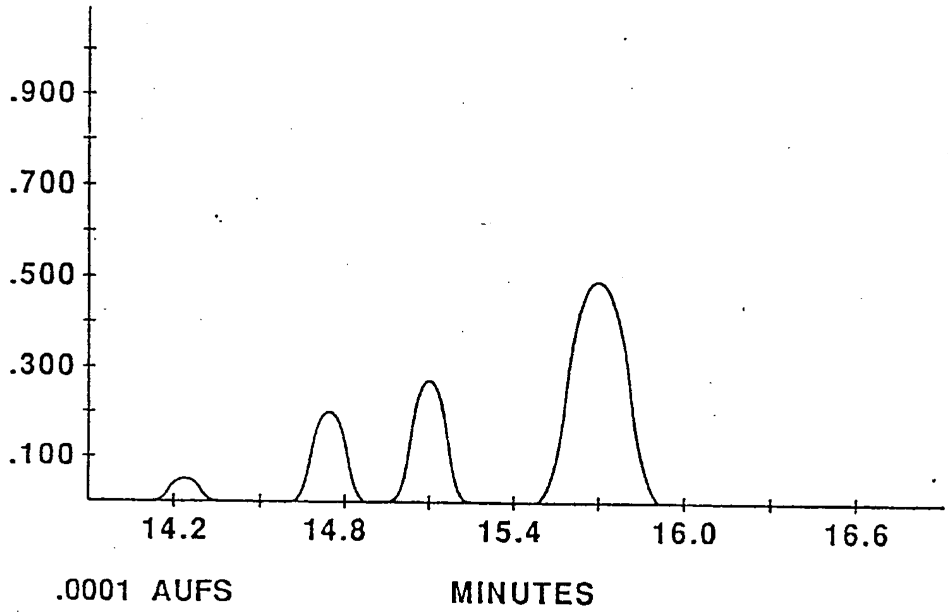

.900
.700
.500
.300
.100
14.2
14.8
15.4
16.0
16.6
.0001 AUFS
MINUTES

Fig. 3M

Fig. 4A

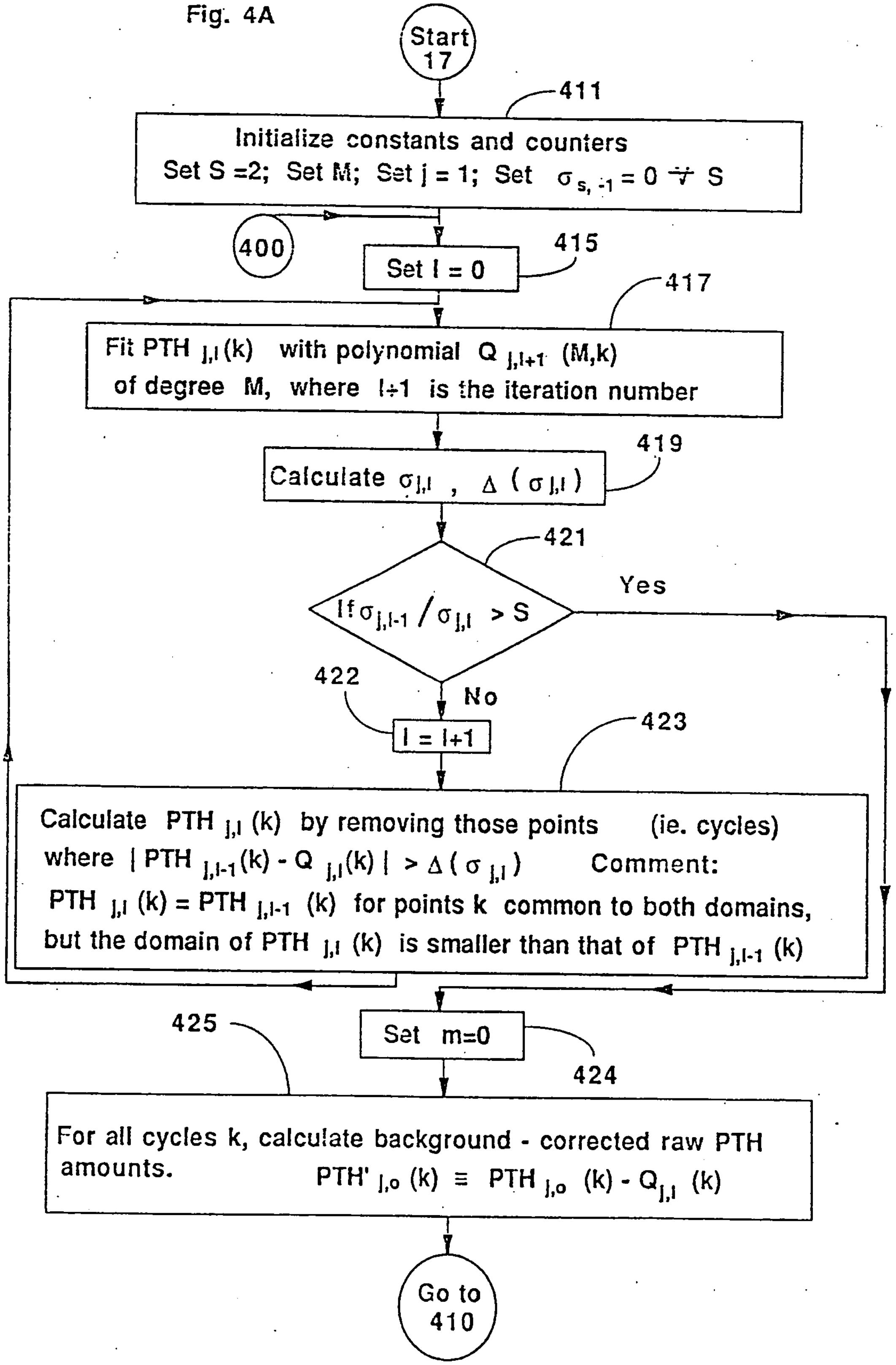

Start 17
411
Initialize constants and counters
Set S =2; Set M; Set j = 1; Set $\sigma_{s,-1} = 0 \forall S$
400
Set l = 0
415
417
Fit PTH $_{j,l}$(k) with polynomial Q $_{j,l+1}$ (M,k) of degree M, where l+1 is the iteration number
419
Calculate $\sigma_{j,l}$, $\Delta(\sigma_{j,l})$
421
If $\sigma_{j,l-1} / \sigma_{j,l}$ > S
Yes
No
422
l = l+1
423
Calculate PTH $_{j,l}$ (k) by removing those points (ie. cycles) where | PTH $_{j,l-1}$(k) - Q $_{j,l}$(k) | > $\Delta(\sigma_{j,l})$ Comment: PTH $_{j,l}$ (k) = PTH $_{j,l-1}$ (k) for points k common to both domains, but the domain of PTH $_{j,l}$ (k) is smaller than that of PTH $_{j,l-1}$ (k)
425
Set m=0
424
For all cycles k, calculate background - corrected raw PTH amounts. PTH' $_{j,o}$ (k) ≡ PTH $_{j,o}$ (k) - Q$_{j,l}$ (k)
Go to 410

Fig. 4B

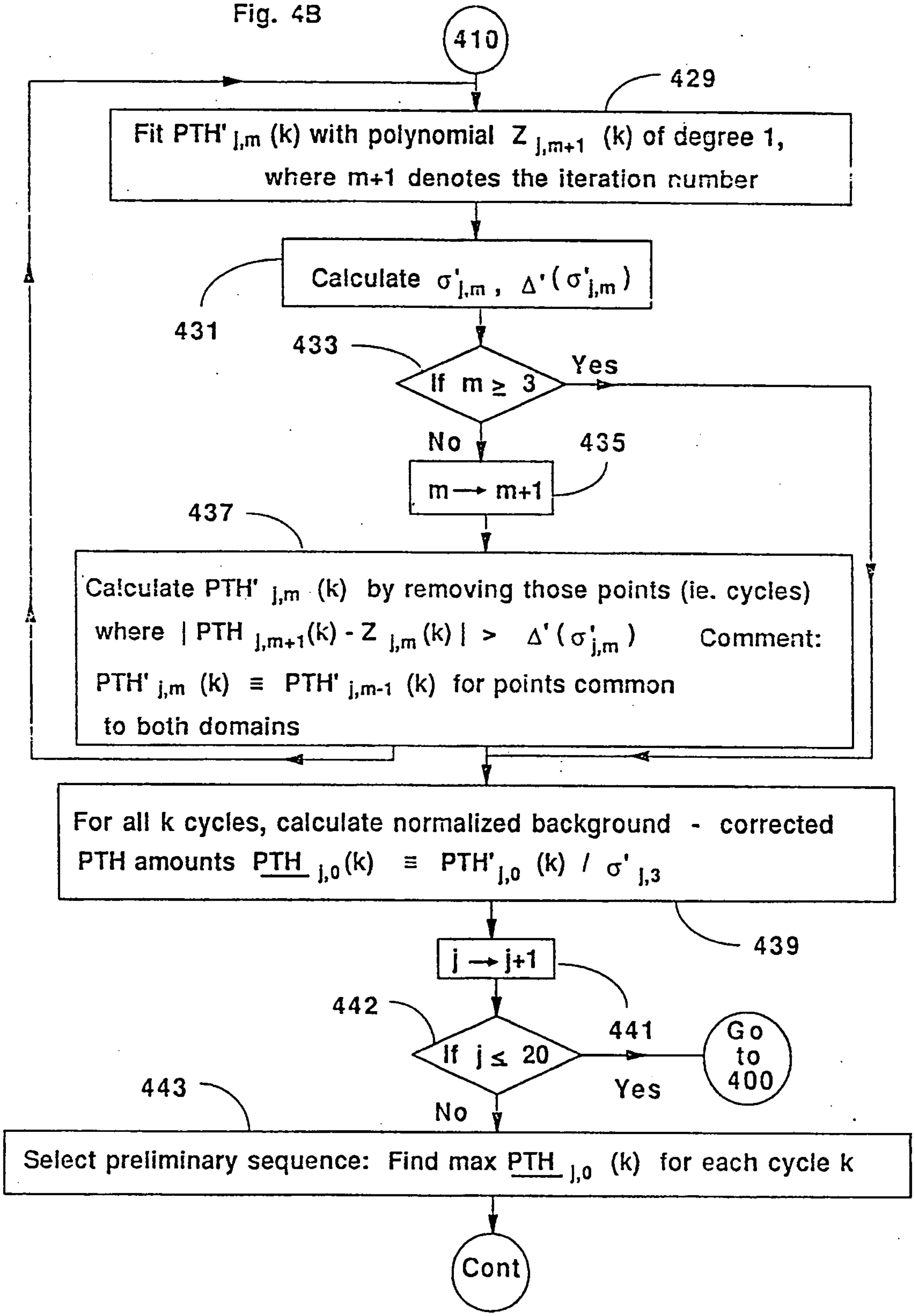

410
429
Fit $PTH'_{j,m}$ (k) with polynomial $Z_{j,m+1}$ (k) of degree 1, where m+1 denotes the iteration number
Calculate $\sigma'_{j,m}$, $\Delta'(\sigma'_{j,m})$
431
433
If $m \geq 3$
Yes
No
435
$m \rightarrow m+1$
437
Calculate $PTH'_{j,m}$ (k) by removing those points (ie. cycles) where $| PTH_{j,m+1}(k) - Z_{j,m}(k) | > \Delta'(\sigma'_{j,m})$ Comment: $PTH'_{j,m}$ (k) $\equiv$ $PTH'_{j,m-1}$ (k) for points common to both domains
For all k cycles, calculate normalized background - corrected PTH amounts $\underline{PTH}_{j,0}(k) \equiv PTH'_{j,0}(k) / \sigma'_{j,3}$
439
$j \rightarrow j+1$
442
441
If $j \leq 20$
Yes
Go to 400
No
443
Select preliminary sequence: Find max $\underline{PTH}_{j,0}$ (k) for each cycle k
Cont

Fig. 5A (Lag correction)

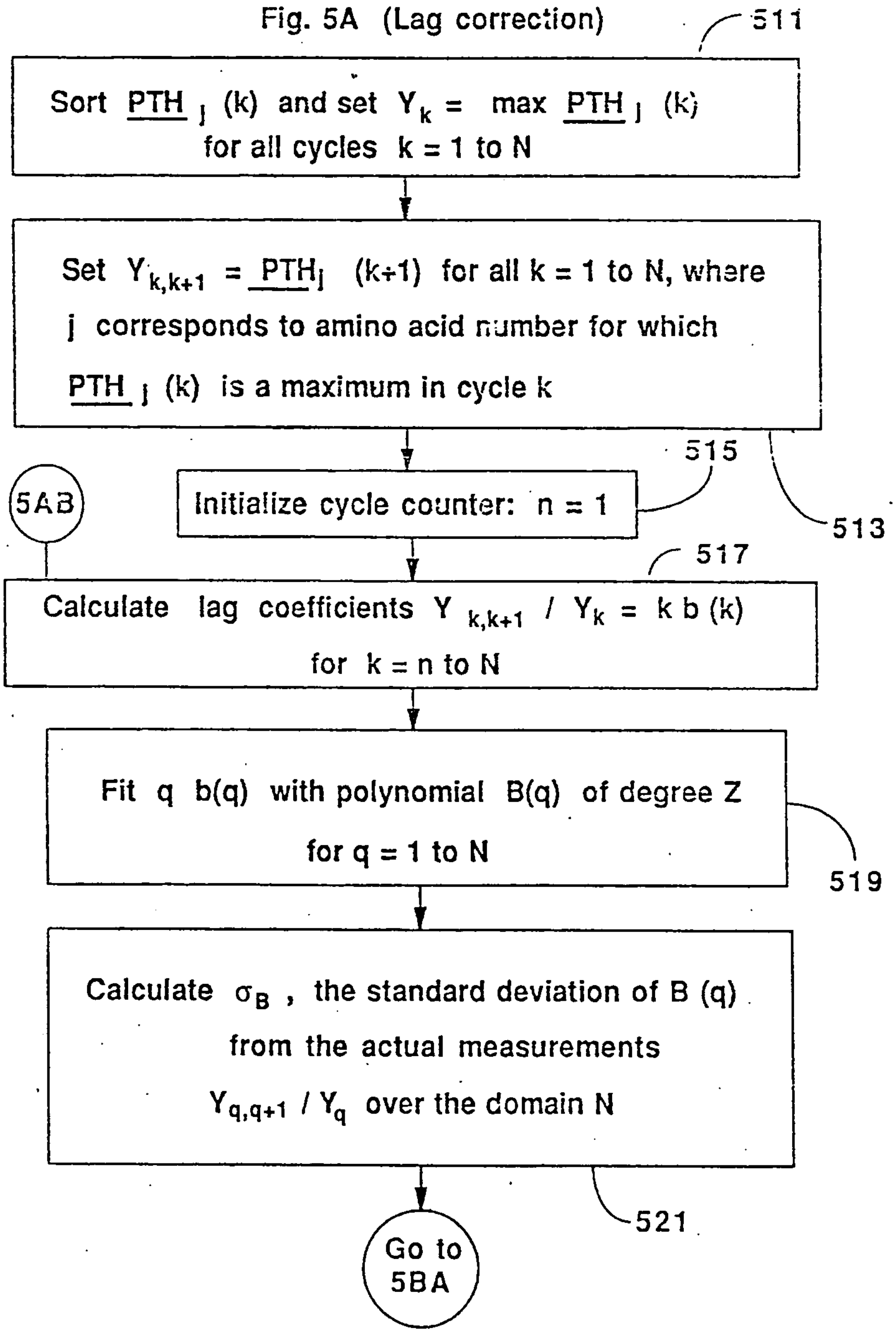

511
Sort $\underline{PTH}_j$ (k) and set Y_k = max $\underline{PTH}_j$ (k) for all cycles k = 1 to N
Set $Y_{k,k+1}$ = $\underline{PTH}_j$ (k+1) for all k = 1 to N, where j corresponds to amino acid number for which $\underline{PTH}_j$ (k) is a maximum in cycle k
513
515
5AB
Initialize cycle counter: n = 1
517
Calculate lag coefficients $Y_{k,k+1}$ / Y_k = k b (k) for k = n to N
Fit q b(q) with polynomial B(q) of degree Z for q = 1 to N
519
Calculate σ_B, the standard deviation of B (q) from the actual measurements $Y_{q,q+1}$ / Y_q over the domain N
521
Go to 5BA

Fig. 5B

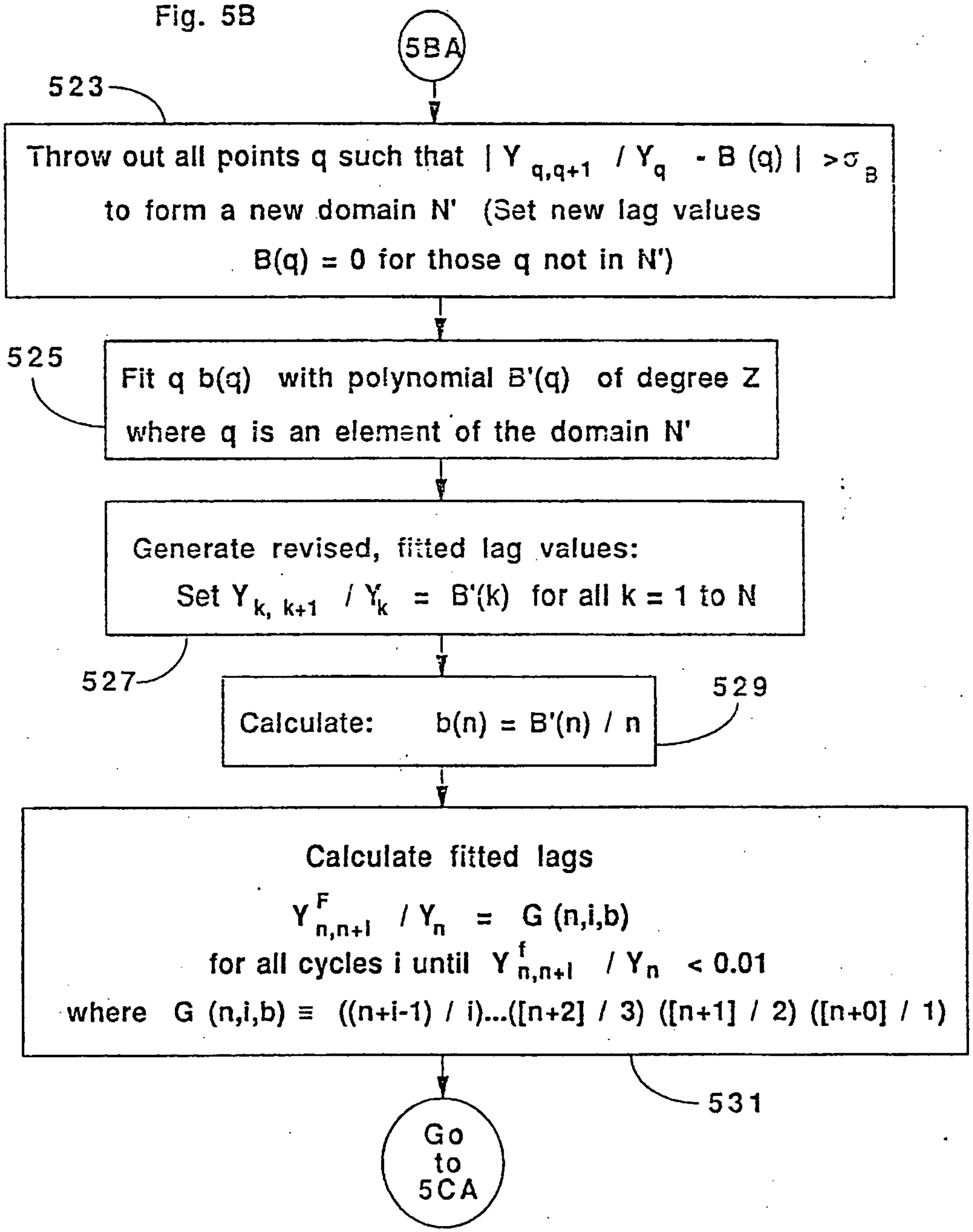

5BA
523
Throw out all points q such that $| Y_{q,q+1} / Y_q - B(q) | > \sigma_B$
to form a new domain N' (Set new lag values
B(q) = 0 for those q not in N')
525
Fit q b(q) with polynomial B'(q) of degree Z
where q is an element of the domain N'
Generate revised, fitted lag values:
Set $Y_{k,\ k+1} / Y_k = B'(k)$ for all k = 1 to N
527
Calculate: b(n) = B'(n) / n
529
Calculate fitted lags
$Y^F_{n,n+l} / Y_n = G(n,i,b)$
for all cycles i until $Y^f_{n,n+l} / Y_n < 0.01$
where $G(n,i,b) \equiv ((n+i-1) / i)...([n+2] / 3)([n+1] / 2)([n+0] / 1)$
531
Go to 5CA

Fig. 5C

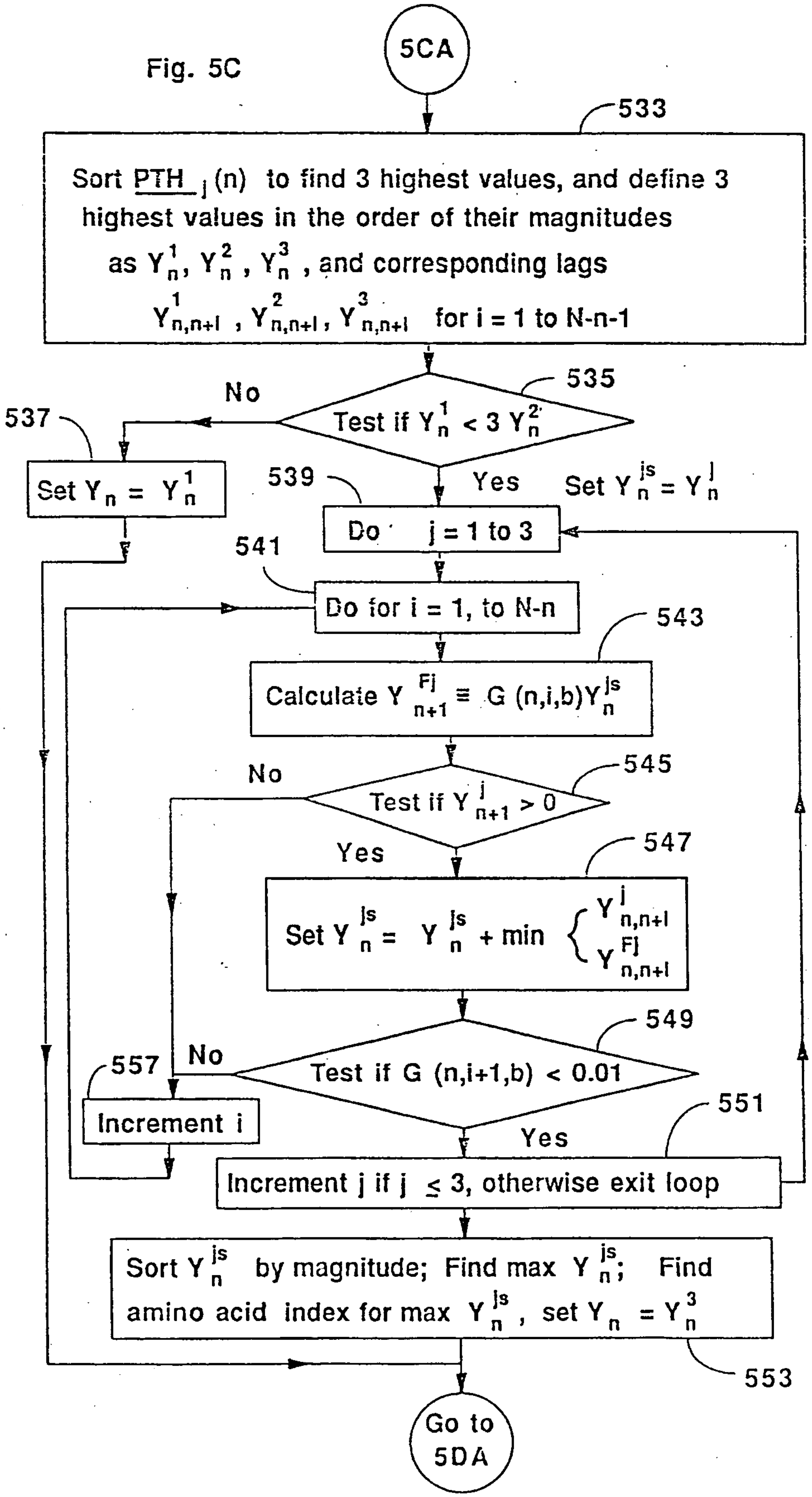

5CA
533
Sort $\underline{PTH}_j(n)$ to find 3 highest values, and define 3 highest values in the order of their magnitudes as Y_n^1, Y_n^2, Y_n^3, and corresponding lags $Y_{n,n+l}^1$, $Y_{n,n+l}^2$, $Y_{n,n+l}^3$ for i = 1 to N-n-1
535
Test if $Y_n^1 < 3\, Y_n^2$
No
537
Set $Y_n = Y_n^1$
Yes
Set $Y_n^{js} = Y_n^j$
539
Do j = 1 to 3
541
Do for i = 1, to N-n
543
Calculate $Y_{n+1}^{Fj} \equiv G(n,i,b) Y_n^{js}$
545
Test if $Y_{n+1}^j > 0$
No
Yes
547
Set $Y_n^{js} = Y_n^{js} + \min \left\{ \begin{matrix} Y_{n,n+l}^j \\ Y_{n,n+l}^{Fj} \end{matrix} \right.$
549
Test if G (n,i+1,b) < 0.01
No
557
Increment i
Yes
551
Increment j if j ≤ 3, otherwise exit loop
Sort Y_n^{js} by magnitude; Find max Y_n^{js}; Find amino acid index for max Y_n^{js}, set $Y_n = Y_n^3$
553
Go to 5DA

Fig. 5D

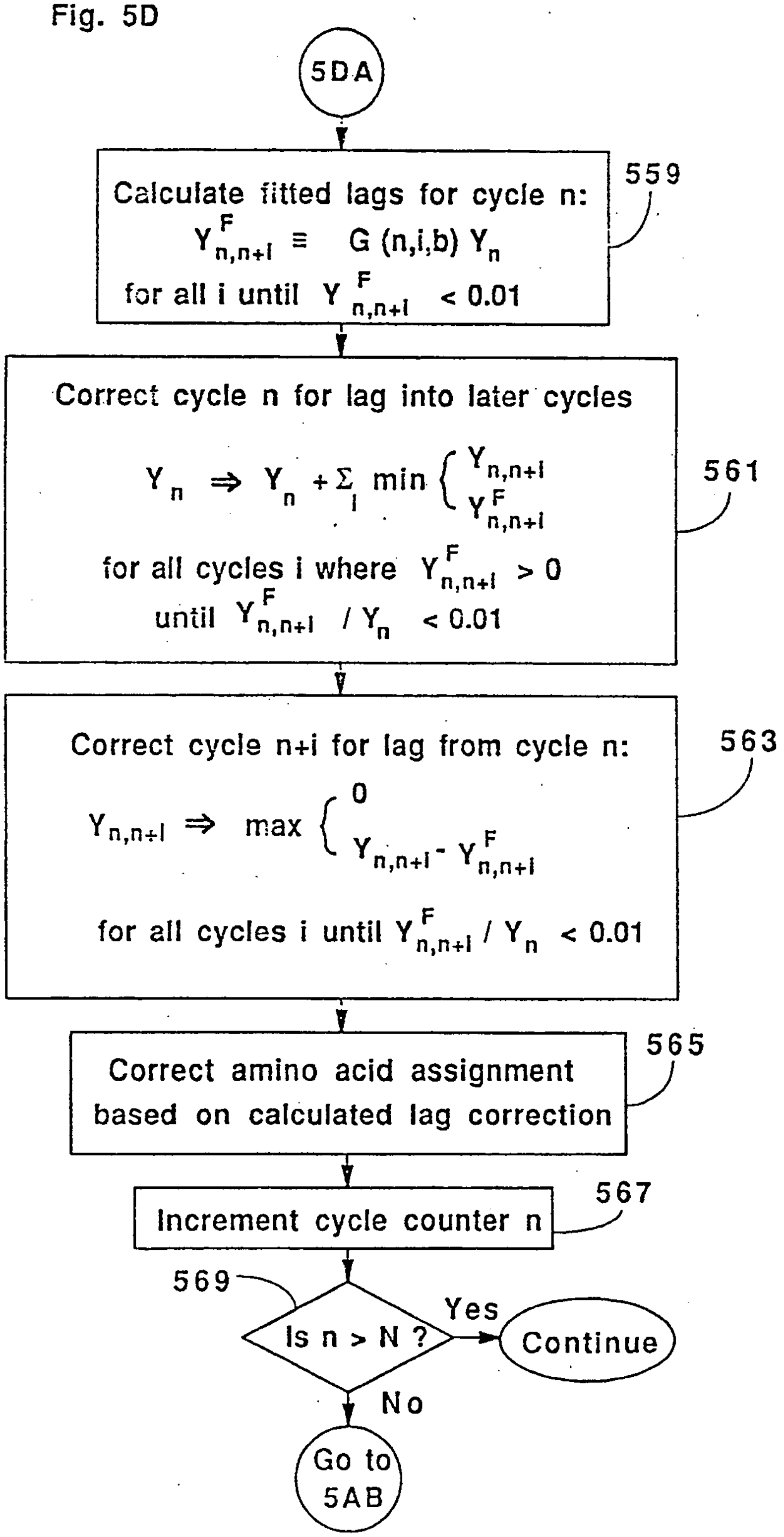
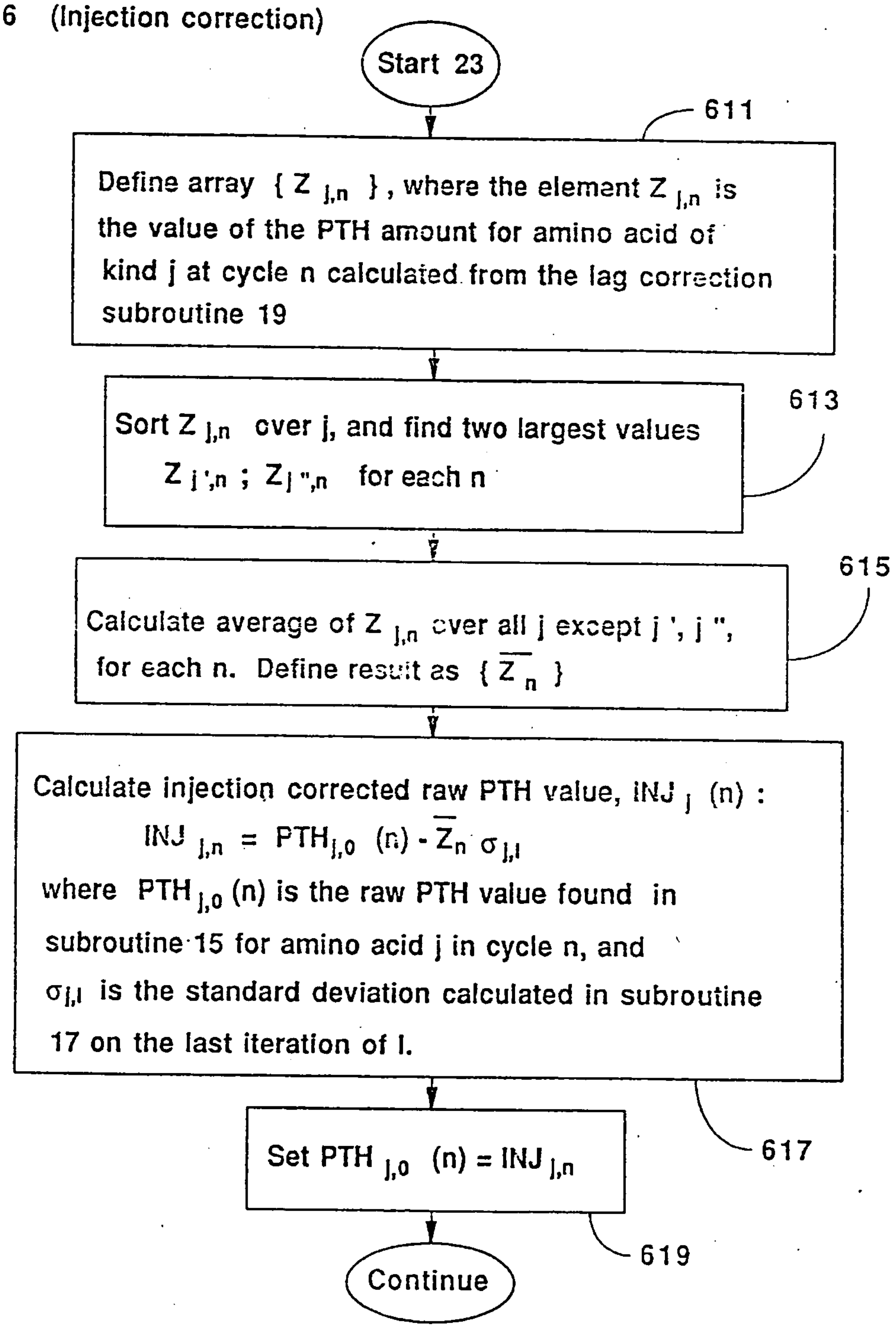
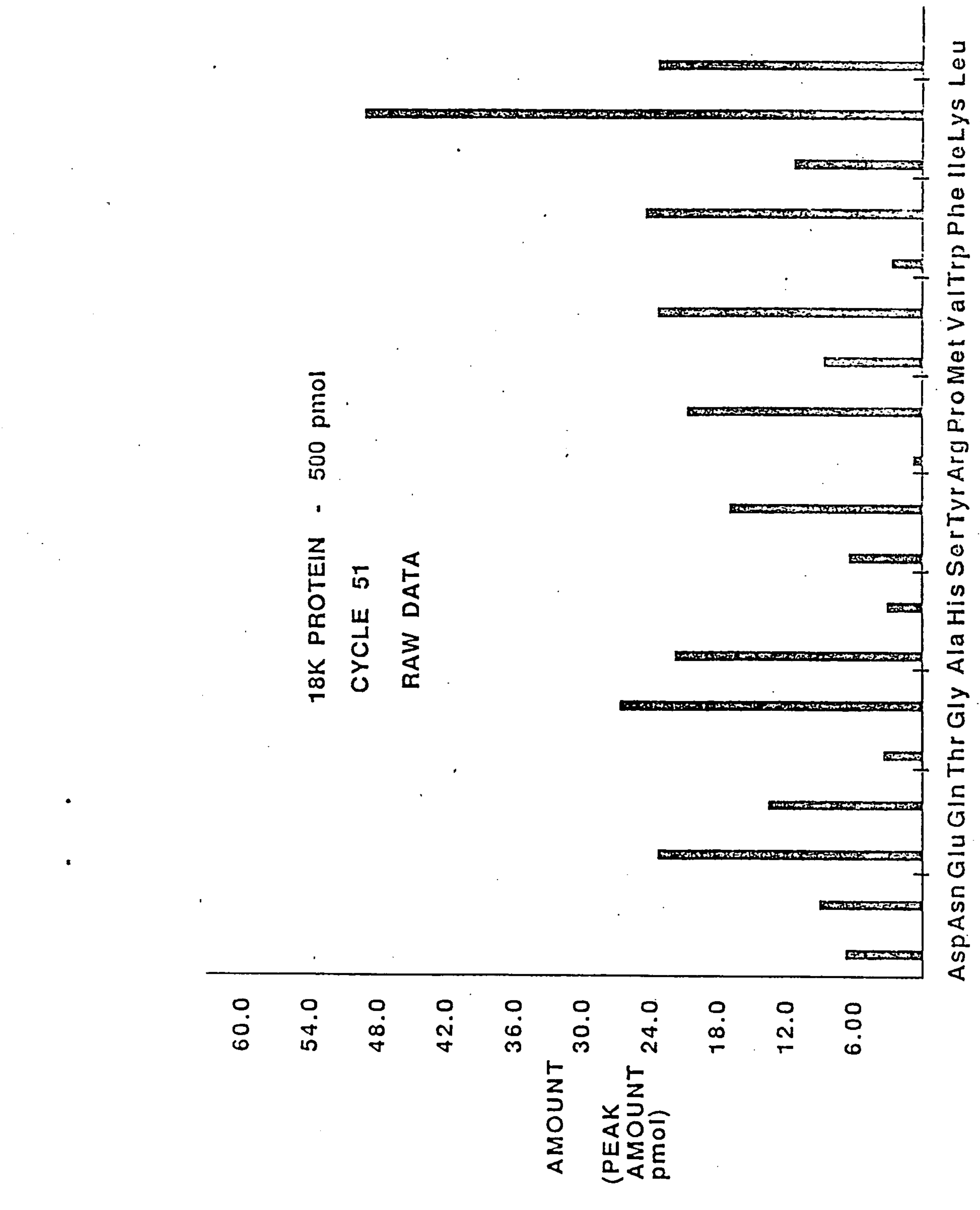

5DA
Calculate fitted lags for cycle n:
$Y^F_{n,n+l} \equiv G(n,l,b)\,Y_n$
for all i until $Y^F_{n,n+l} < 0.01$
559
Correct cycle n for lag into later cycles
$Y_n \Rightarrow Y_n + \sum_l \min \begin{cases} Y_{n,n+l} \\ Y^F_{n,n+l} \end{cases}$
for all cycles i where $Y^F_{n,n+l} > 0$
until $Y^F_{n,n+l} / Y_n < 0.01$
561
Correct cycle n+i for lag from cycle n:
$Y_{n,n+l} \Rightarrow \max \begin{cases} 0 \\ Y_{n,n+l} - Y^F_{n,n+l} \end{cases}$
for all cycles i until $Y^F_{n,n+l} / Y_n < 0.01$
563
Correct amino acid assignment based on calculated lag correction
565
Increment cycle counter n
567
569
Is n > N ?
Yes
Continue
No
Go to 5AB

Fig. 6 (Injection correction)


Start 23
611
Define array $\{ Z_{j,n} \}$, where the element $Z_{j,n}$ is the value of the PTH amount for amino acid of kind j at cycle n calculated from the lag correction subroutine 19
613
Sort $Z_{j,n}$ over j, and find two largest values $Z_{j',n}$; $Z_{j'',n}$ for each n
615
Calculate average of $Z_{j,n}$ over all j except j ', j ", for each n. Define result as $\{ \overline{Z}_n \}$
Calculate injection corrected raw PTH value, INJ_j (n) :
$INJ_{j,n} = PTH_{j,0} (n) - \overline{Z}_n \sigma_{j,l}$
where $PTH_{j,0}$ (n) is the raw PTH value found in subroutine 15 for amino acid j in cycle n, and $\sigma_{j,l}$ is the standard deviation calculated in subroutine 17 on the last iteration of l.
617
Set $PTH_{j,0}$ (n) = $INJ_{j,n}$
619
Continue



60.00 FULL SCALE

Fig. 7A

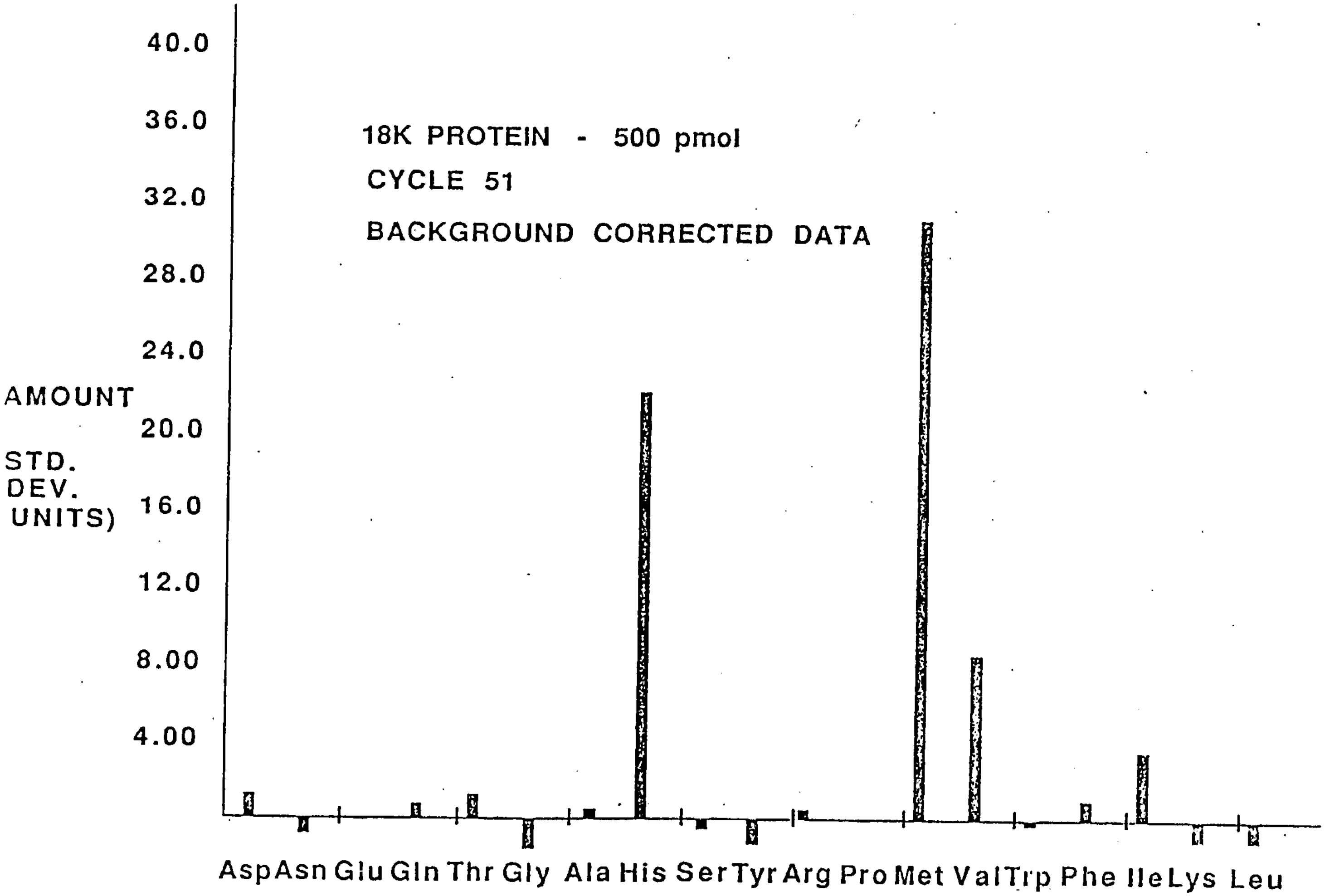

18K PROTEIN - 500 pmol
CYCLE 51
BACKGROUND CORRECTED DATA
AMOUNT
STD.
DEV.
UNITS)
40.0
36.0
32.0
28.0
24.0
20.0
16.0
12.0
8.00
4.00
Asp Asn Glu Gln Thr Gly Ala His Ser Tyr Arg Pro Met Val Trp Phe Ile Lys Leu
40.00 FULL SCALE

Fig. 7B

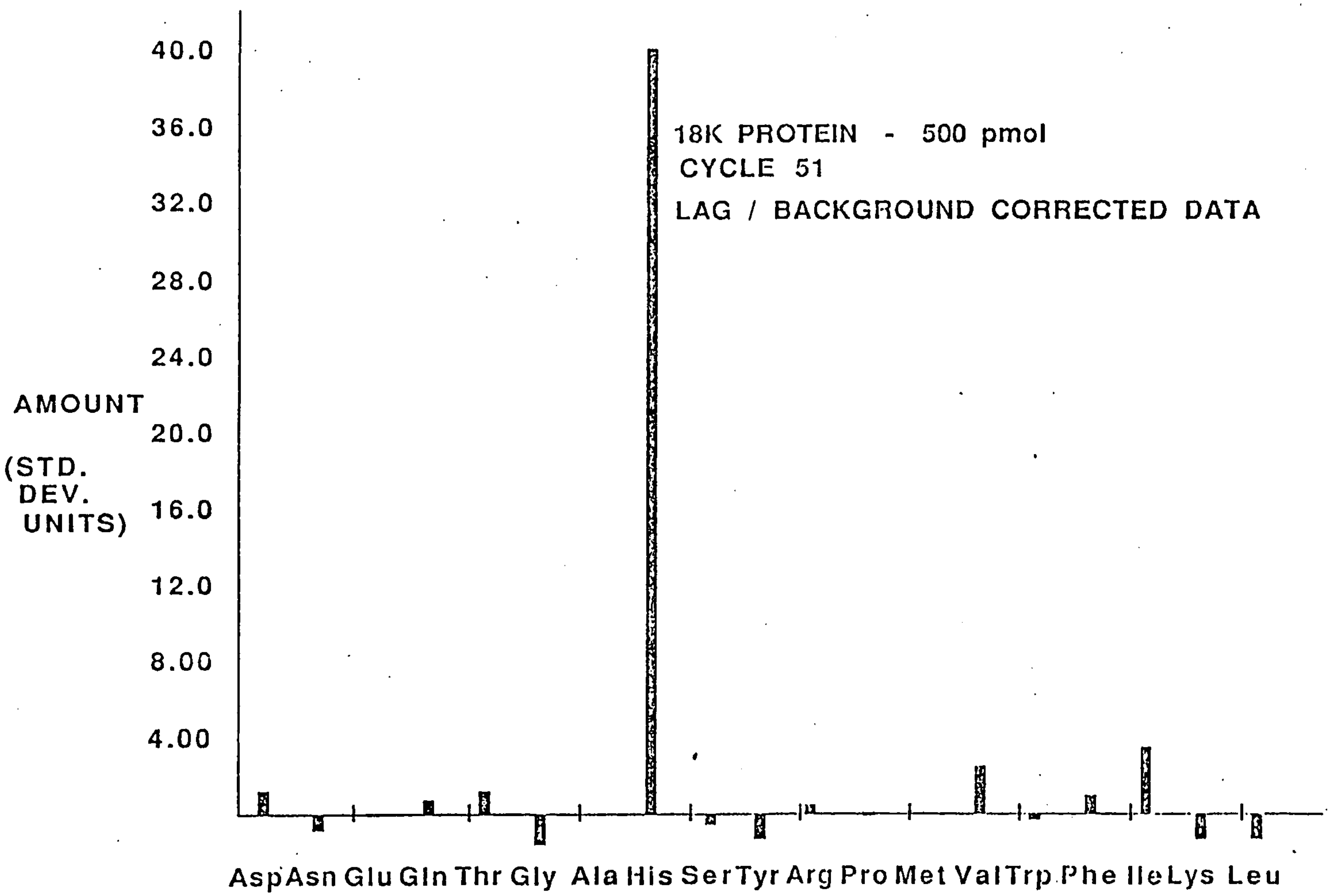

40.0
36.0
32.0
28.0
24.0
AMOUNT
20.0
(STD.
DEV.
UNITS)
16.0
12.0
8.00
4.00
18K PROTEIN - 500 pmol
CYCLE 51
LAG / BACKGROUND CORRECTED DATA
Asp Asn Glu Gln Thr Gly Ala His Ser Tyr Arg Pro Met Val Trp Phe Ile Lys Leu
40.00 FULL SCALE

Fig. 7C

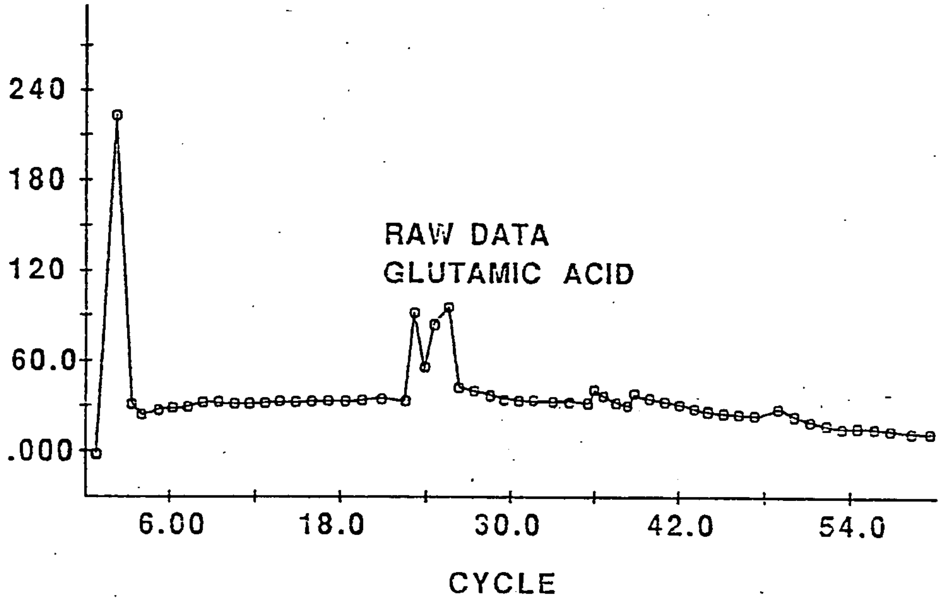

240
180
120
60.0
.000
6.00
18.0
30.0
42.0
54.0
CYCLE
RAW DATA
GLUTAMIC ACID

Fig. 8A

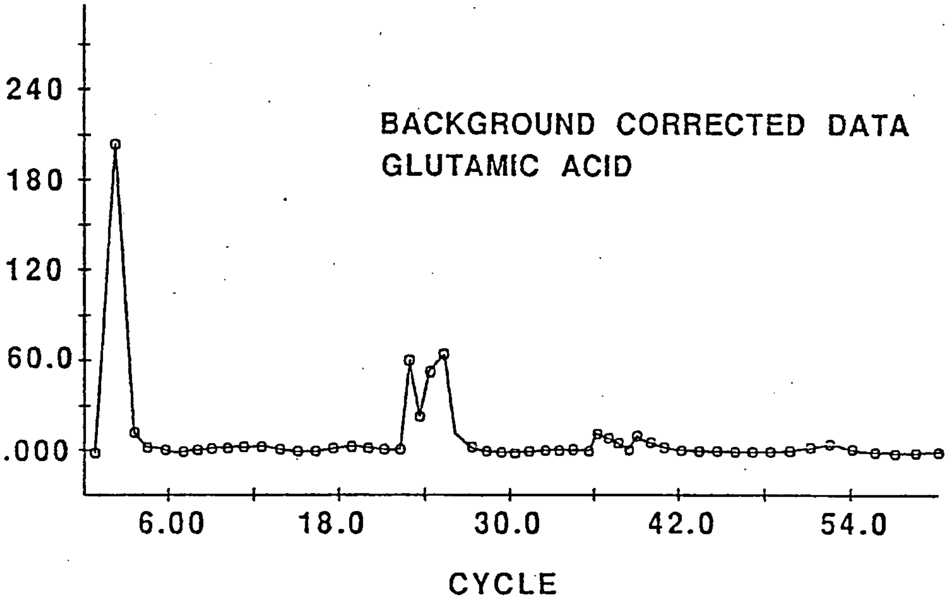

240
180
120
60.0
.000
6.00
18.0
30.0
42.0
54.0
CYCLE
BACKGROUND CORRECTED DATA
GLUTAMIC ACID

Fig. 8B

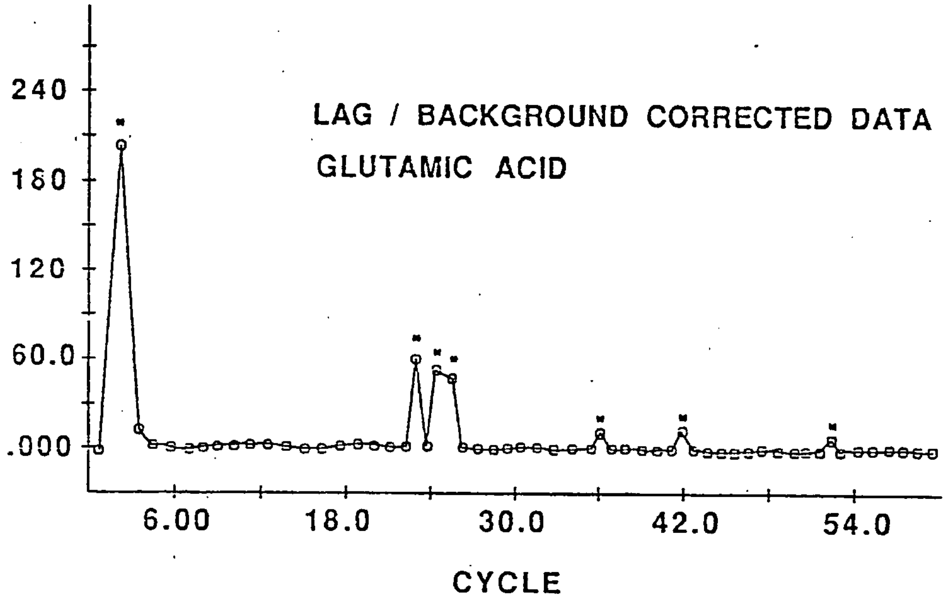

LAG / BACKGROUND CORRECTED DATA
GLUTAMIC ACID
240
180
120
60.0
.000
6.00
18.0
30.0
42.0
54.0
CYCLE

Fig. 8C